NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A Study of the Diffraction Behavior
and Resolution Criteria for Pattern Recognition
for a Proposed Multiplexed Imaging Technique

by

Brian Jay Musselman
SEPTEMBER 1991

Thesis Advisor D.S. Davis

Approved for public release; Distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (*If applicable*) PH/Dv	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (*City, State, and ZIP Code*) Monterey, Ca 93943-5000	7b ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (*If applicable*)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (*City, State, and ZIP Code*)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (*Include Security Classification*)
"A Study of the Diffraction Behavior and Resolution Criteria for Pattern Recognition for a Proposed Multiplexed Imaging Technique"

12 PERSONAL AUTHOR(S)
Brian Jay Musselman

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (*Year, Month, Day*)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	September 1991	172

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the U.S. Government

17 COSATI CODES			18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	Walsh Function Encoding Masks; Multiplexed Imaging; Pattern Recognition; Sequency Theory

19 ABSTRACT (*Continue on reverse if necessary and identify by block number*)

This project quantifies several aspects of a new multiplexed imaging technique proposed by D.S. Davis. The novel approach of this technique involves the use of encoding masks derived from a basis set of two-dimensional Walsh functions. There were two distinct problems addressed by this thesis research. First, a study of computer simulated diffraction patterns of the photon flux through these encoding masks yielded design constraints to be incorporated into a prototype system. These constraints were expressed in a simple mathematical relation in terms of wavelength, diffraction angle, and spatial frequency. A second problem addressed the minimum spatial resolution required for pattern recognition. The conclusion reached was that the minimum number of resolution elements necessary for pattern recognition is 64 in each direction. This determination also fixed the minimum size of the basis set of two-dimensional Walsh functions required for multiplexing, as well as the number of pixels required to display the image.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL D. S. Davis	22b TELEPHONE (*Include Area Code*) (408) 646-2877 22c OFFICE SYMBOL PH/Dv

Approved for public release: Distribution is unlimited

A Study of the Diffraction Behavior and Resolution Criteria for Pattern Recognition for a Proposed Multiplexed Imaging Technique

by

Brian Jay Musselman
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
SEPTEMBER 1991

ABSTRACT

This project quantifies several aspects of a new multiplexed imaging technique proposed by D.S. Davis. The novel approach of this technique involves the use of encoding masks derived from a basis set of two-dimensional Walsh functions. There were two distinct problems addressed by this thesis research. First, a study of computer simulated diffraction patterns of the photon flux through these encoding masks yielded design constraints to be incorporated into a prototype system. These constraints were expressed in a simple mathematical relation in terms of wavelength, diffraction angle, and spatial frequency. A second problem addressed the minimum spatial resolution required for pattern recognition. The conclusion reached was that the minimum number of resolution elements necessary for pattern recognition is 64 in each direction. This determination also fixed the minimum size of the basis set of two-dimensional Walsh functions required for multiplexing, as well as the number of pixels required to display the image.

TABLE OF CONTENTS

I. INTRODUCTION

A. MOTIVATION

This thesis is part of a much larger research and development effort being conducted by the author's thesis advisor [Ref. 1]. The ultimate goal of this larger project is to develop and evaluate a new type of instrument for infrared imaging and imaging spectroscopy. The key feature of this new technique is that it will multiplex images, by means of focal plane encoding masks, onto a single detector, or at most a pair of detectors. The viability of this method was demonstrated in the proof-of-concept thesis research conducted by Capt R.H. McKenzie, III, USMC [Ref. 2]. The motives for this thesis project have been twofold. First, since the masking technique will be employed at long wavelengths, a quantitative prediction of the diffraction behavior is needed before mask design is undertaken. This project successfully addresses that problem. The second motive for this research project is a direct consequence of the multiplexing technique itself. The spatial resolution of a multiplexed image will be inversely proportional to the number of encoding masks incorporated into a working instrument [Ref. 1]. This project, therefore, successfully provides initial estimates for the number of encoding masks needed to produce

recognizable multiplexed images from a variety of commonplace objects or targets.

B. BACKGROUND

Terrestrial objects emit most of their thermal radiation at infrared (IR) wavelengths. Many interesting objects are in approximate thermal equilibrium with their surroundings, which have temperatures in the 275 - 310 K range. Assuming that these sources radiate approximately as blackbodies [Ref. 3], the bulk of their radiance lies within the 5 - 25 μm wavelength range. Of course, warmer, man-made objects will radiate at correspondingly shorter wavelengths. It is desirable to utilize this abundance of radiated energy to detect and to identify objects. Infrared images provide information about the size, shape, and temperature of the radiating sources. Infrared spectra reveal the chemical composition and thermodynamic information.

D.S. Davis, advisor on this thesis project, has been involved with state-of-the-art spectroscopic IR instrumentation for many years. Historically, his research interests have involved the development and use of unique, high resolution, fully multiplexed, infrared Fourier transform spectrometers. His interest in IR multiplexing techniques inspired the concept of multiplexed imaging, which is one of the goals of the larger research project of which this thesis is a part. [Ref. 1]

The purpose of infrared imaging is to measure the image's irradiance as a function of two independent spatial variables, such as conventional Cartesian (x,y) coordinates. Spectroscopic imaging introduces a third independent variable, such as wavelength or frequency. Traditional IR imaging devices and imaging spectrometers typically scan over one or more of the independent variables which describe image irradiance. There is, however, an inherent inefficiency associated with such scanning techniques. This inefficiency stems from the fact that, by observing only one small element of the image field (and/or one spectral element) at a time, the instrument ignores most of the energy from the scene. This can have severe consequences in the infrared, because available photons from a target are precious, and they typically comprise only a small fraction of the entire, background-dominated, IR irradiance. The goal of the proposed technique is to develop a more efficient generation of instruments which will apply the proven advantages of infrared Fourier transform spectroscopy (FTS) to infrared imaging and infrared spectroscopy. [Ref. 1]

C. STATEMENT OF THESIS PROBLEM

This thesis research addresses two fundamental questions which must be answered before the design of a prototype multiplexed imaging instrument is complete. First, since the instrument will operate at comparatively long wavelengths,

there is concern that diffraction effects produced by the encoding masks will cause substantial beam spreading and degradation of image resolution. Therefore, the initial portion of this thesis project quantifies the diffraction behavior of the masks, so that this behavior can be included in subsequent instrumental design considerations. Second, the image spatial resolution required to produce identifiable images of ordinary terrestrial objects must be determined. When the proposed encoding scheme operates at short wavelengths (far removed from the diffraction limits considered in the first part of the research), it is conceptually capable of extremely high spatial resolution. In fact, the resolution will be inversely proportional to the number of encoding masks used. Determination of the minimum useful number of masks is important so that needless effort and resources are not wasted on an over-designed prototype instrument.

II. OVERVIEW OF MULTIPLEXED IMAGING

This section provides a synopsis of the multiplexed imaging technique. Although the ultimate goal of the entire project is to develop a multiplexed imaging spectrometer, the thesis research at hand considers only multiplexed imaging. Therefore, the spectroscopic aspects of the larger project are ignored.

A. MULTIPLEXING CONCEPTS

The key feature of optical multiplexing, and the one that produces its inherent efficiency, is that all of the photons from the target are detected all of the time. This is achieved in a multiplexed scanning instrument by detecting linear combinations of the signals of interest, rather than by scanning through sets of the individual signals themselves. For instance, consider the operation of a raster scanning imager, such as a FLIR. Let ϕ_i be the radiative flux through the ith pixel of the image, and P_i be the instantaneous power sensed by the detector. Assume that there are no inefficiencies or losses associated with the detection process. Then, for N pixels which comprise the entire image,

$$P_i = \phi_i \ , \ \{i = 1,\ldots,N\} \ . \tag{1}$$

While the device observes the ith pixel, it ignores the other N - 1 pixels.

A multiplexed sensor detects a linear combination of the pixel fluxes

$$P_i = \sum_{j=1}^{N} \mu_{ij}\phi_j \ , \ \{i = 1,\ldots,N\} \ , \tag{2}$$

where the μ_{ij} are weighting coefficients. Therefore, one scans through N μ's rather than N ϕ's. The multiplexing weights μ_{ij} are not chosen arbitrarily; there are several constraints which they must meet [Ref. 1]. First, equation (2) must be invertible, so that the individual pixel fluxes ϕ_j can be recovered from the measurements of the P_i. Second, the μ_{ij} need to permit as much radiation as possible to pass to the detector, in order to maximize the overall optical efficiency. The inherent inefficiency of traditional raster scan devices is due to the fact that their

$$\mu_{ij} = \delta_{ij} \ , \tag{3}$$

where δ_{ij} is the usual Kronecker delta symbol. Third, the μ_{ij} must be capable of being implemented in a practical optical design which is suitable for a working instrument.

For example, consider the process by which a Fourier transform spectrometer (FTS) encodes spectral intensities [Ref. 1]. In this instance, the ϕ_j represent spectral fluxes at N different frequencies or wavelengths. The μ_{ij} are discrete samples of the familiar orthogonal sinusoidal

functions of Fourier analysis. The P_i are the discrete samples of the recorded interferogram. Inversion of equation (2) to yield the spectral fluxes is accomplished by applying a discrete Fourier transform to the P_i.

B. SEQUENCY THEORY AND MULTIPLEXED IMAGING

The family of functions chosen to implement the encoding scheme for this multiplexed imaging project are the Walsh functions from sequency theory [Refs. 1,4] rather than the sinusoidal functions associated with FTS and Fourier analysis. The Walsh functions are similar to the sinusoids in that they constitute complete, orthogonal basis sets. In other respects, however, they are radically different. The most striking differences are that they are discontinuous, and that they assume only one of two possible values, ± 1. Another difference is that the Walsh functions are not necessarily periodic. Furthermore, a complete basis set of Walsh functions must contain $N = 2^m$ members, where m is a nonnegative integer. Detailed discussions of these and other properties may be found in Beauchamp [Ref. 4].

The concept of a Walsh function's sequency is fundamental to understanding this thesis research. Its definition is straightforward. The domain over which a basis set of Walsh functions is defined is partitioned into N subdomains of equal length, where N is the number of functions in the complete basis set. The Walsh functions then assume constant values,

either +1 or -1, across each subdomain, according to the rules which dictate their generation [Ref. 4]. At the boundaries between adjacent subdomains, a function may switch values, thereby crossing zero. The total number of such zero crossings that occur across the entire domain of the function's definition is that function's sequency. For a complete Walsh basis set, the corresponding member function's sequencies range from zero to N - 1. N is referred to as the order of the basis set.

C. THE NEW IMAGE ENCODING TECHNIQUE

The new multiplexed image encoding technique will utilize Walsh functions to generate optical encoding masks. These masks will, in turn, be used to modulate the flux distribution in an image, in analogy with equation (2). An illustration of how this might be done is provided by the following example. Suppose that one desires to determine the radiant fluxes ϕ_j through different contiguous regions of an image field. Consider the complete Walsh basis of order 4. Construct a one-by-four row matrix for each basis function in the set, such that each function's subdomain structure corresponds to one entry, or matrix element:

Sequency	Row Matrix
0	(+1 +1 +1 +1)
1	(-1 -1 +1 +1)
2	(-1 +1 +1 -1)
3	(+1 -1 +1 -1)

The reader may easily verify that the rows are mutually orthogonal; in fact, so are the columns. Hence, the rows could be combined into a single four-by-four orthogonal Walsh encoding matrix, such that

$$\mu_{ij} = \begin{pmatrix} +1 & +1 & +1 & +1 \\ -1 & -1 & +1 & +1 \\ -1 & +1 & +1 & -1 \\ +1 & -1 & +1 & -1 \end{pmatrix}. \tag{4}$$

The practical optical implementation of this matrix would involve construction of four masks, each of which is portioned into four regions where boundaries match those of the image regions to be multiplexed. A possible example of such a portioning is shown in Figure 1. Each mask in Figure 1 corresponds to one row of matrix (4). Dark regions imply a matrix element of +1 and signify that the mask is optically transmitting across those pixels. Light regions correspond to matrix elements of -1 and denote that those pixels are optically reflective. Light that is transmitted is collected and sent to one detector; that which is reflected is sent to

a second detector. The second detector's output is subtracted from that of the first detector. As the encoding instrument cycles through the four masks, the differenced detector signal will be proportional to the P_i in equation (2). Collecting the four P_i's into a single column vector and multiplying it by the inverse of μ_{ij} would yield the desired ϕ_j's.

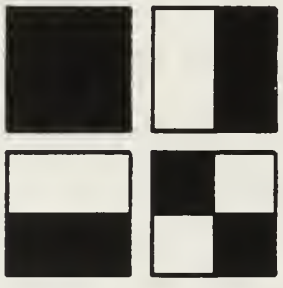

Figure 1 Two-dimensional Sequency -Ordered Walsh Masks for Four Pixels

The actual encoding scheme in a prototype multiplexed imaging system will not make explicit use of the two-dimensional mask configurations like those in Figure 1. Rather, the prototype will utilize two essentially one-dimensional mask families, and it will exploit a very powerful mathematical feature of the Walsh functions, the Kronecker product [Ref. 4]. Via this approach, a two-dimensional mask basis may be synthesized from two lower-order one dimensional bases. This is again illustrated by example. Consider the Walsh basis of order two. Again, write down the corresponding

matrices, but this time as both row (one-by-two) and column (two-by-one) matrices, as shown below.

$$\begin{array}{c|cc} & (+1 \quad +1) & (-1 \quad +1) \\ \hline \begin{bmatrix} +1 \\ +1 \end{bmatrix} & \begin{bmatrix} +1 & +1 \\ +1 & +1 \end{bmatrix} & \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix} \\ \begin{bmatrix} -1 \\ +1 \end{bmatrix} & \begin{bmatrix} -1 & -1 \\ +1 & +1 \end{bmatrix} & \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \end{array}$$

Multiply together the two numbers located on the outer row-column intersections, and collect the products in a two-by-two groupings, as shown. Note that these groupings, called Kronecker products, are exact numerical analogs of the mask pixel structures in Figure 1. The new multiplexed imaging technique will implement an optical scheme to incorporate this property, thereby greatly reducing the number and complexity of the encoding mask families required [Ref. 1]. Figure 2 illustrates a complete 64 pixel "checkerboard" mask basis, synthesized from two one-dimensional bases of order eight.

D. RESTATEMENT OF THESIS GOALS IN WALSH SEQUENCY-THEORETICAL TERMS

As a consequence of the foregoing discussion, the two goals of this thesis research expressed in section I.C. may be rephrased in sequency-theoretical terms: (1) since the prototype system will generate elaborate two-dimensional image encoding masks via the Kronecker product, the system designers

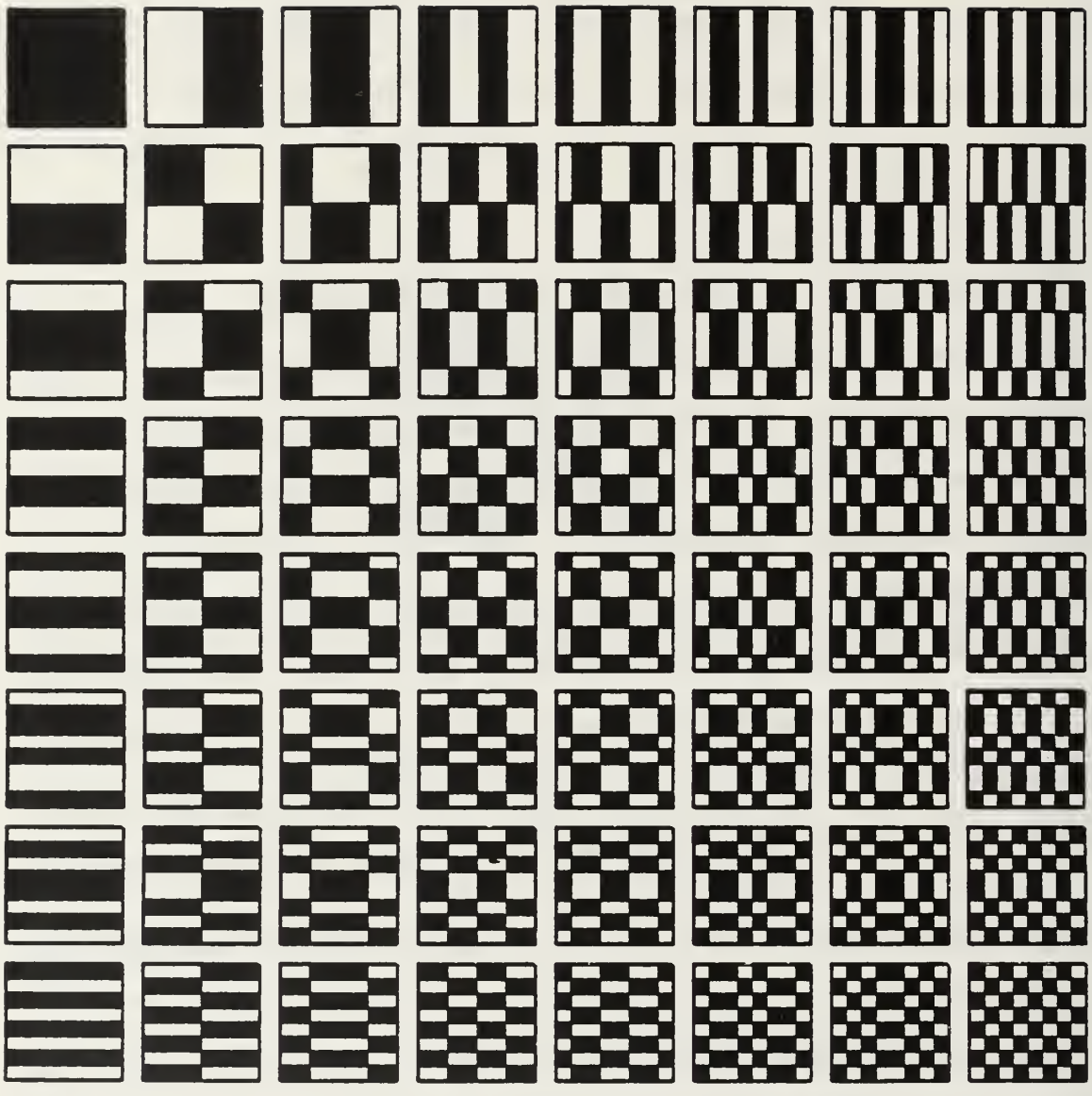

Figure 2 Two-dimensional Sequency-Ordered Walsh Masks for 64 Pixels

need to know how these masks will diffract long wavelength radiation, as well as encode it. Diffraction effects could cause severe beam spreading problems, resulting in substantial loss of optical efficiency, particularly at these longer wavelengths and/or when using encoding masks of high spatial sequency. Therefore, the first goal of this research has been to predict these effects quantitatively. (2) The second goal is stated more succinctly: the system designers need to know how many spatial sequencies are sufficient to encode images which remain recognizable once decoded. It would be both expensive and more optically complex to build a prototype instrument with a far larger Walsh basis set (i.e., containing bases of higher order) than actually needed.

III. THE DIFFRACTION PROBLEM

As noted in the Chapters I and II, one of the basic tasks of this thesis research project was to simulate the behavior of the proposed multiplexed imaging scheme at long wavelengths. Therefore, this first phase of the research is called the diffraction problem. It seeks to quantify the diffraction behavior of the encoding masks by means of computer simulation.

A. BASIC SCALAR DIFFRACTION THEORY

No discussion of an imaging system is complete without considering the effects of interference and diffraction of the incident wavefronts. The importance of these effects cannot be understated.

> "Diffraction effects are accordingly of great significance in the detailed understanding of devices containing lenses, stops, source slits, mirrors, and so on. If all defects in a lens system were removed, the ultimate sharpness of the image would be limited by diffraction." [Ref. 3]

The essential feature of diffraction is that an obstruction deviates the rectilinear propagation of a light wave. The geometry of diffraction through a rectangular aperture is summarized by Hecht [Ref. 3] and depicted in Figure 3. References to the aperture plane are in lower case letters; those to the image plane are in capitals.

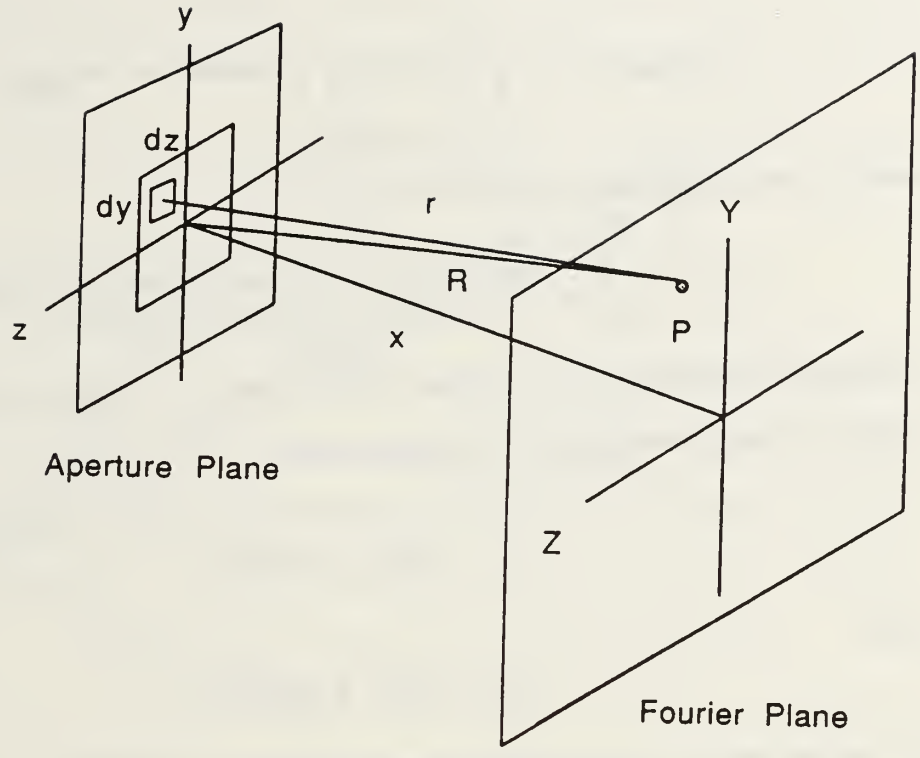

Figure 3 Diffraction Through a Rectangular Aperture

As predicted by the Huygens-Fresnel principle, a differential element of surface area within the aperture, dS (equivalent to dydz), may be considered as comprised of an infinite number of secondary wave sources. However, for a source of monochromatic light, dS is much smaller than the wavelength λ, and all of dS's contributions to the electromagnetic disturbance at point P in the image plane arrive in phase and interfere constructively. Assuming uniform, normally incident illumination over the entire aperture, it follows that

$$E(k_Y,k_Z) \propto \int_{-\infty}^{\infty}\int A(y,z)\, e^{-i\kappa(y\sin\phi + z\sin\theta)}\, dydz \ , \tag{5}$$

where the aperture function is

$$A(y,z) = \begin{cases} 1 & \textit{for } (y,z) \textit{ in transparent regions} \\ 0 & \textit{for } (y,z) \textit{ in opaque regions} \end{cases} . \tag{6}$$

The wave number κ, is $2\pi/\lambda$. The spatial frequencies in the y and z directions are, respectively,

$$k_Y = \kappa Y/R = \kappa\sin\phi \ , \tag{7}$$

and

$$k_Z = \kappa Z/R = \kappa\sin\theta \ . \tag{8}$$

E represents the diffracted electric field distribution. Figure 4 illustrates the angular relationship of ϕ and θ between the aperture and image planes.

Equation (5) indicates that the electromagnetic field distribution at the image plane is proportional to the two-dimensional Fourier transform of the aperture function. Determining the diffraction behavior of the photon flux through the Walsh masks requires a three-step calculation. First, simulate the aperture function for each mask of interest. Second, calculate the two-dimensional Fourier transform of that mask. Third, calculate the actual distribution of the resulting irradiance as a function of

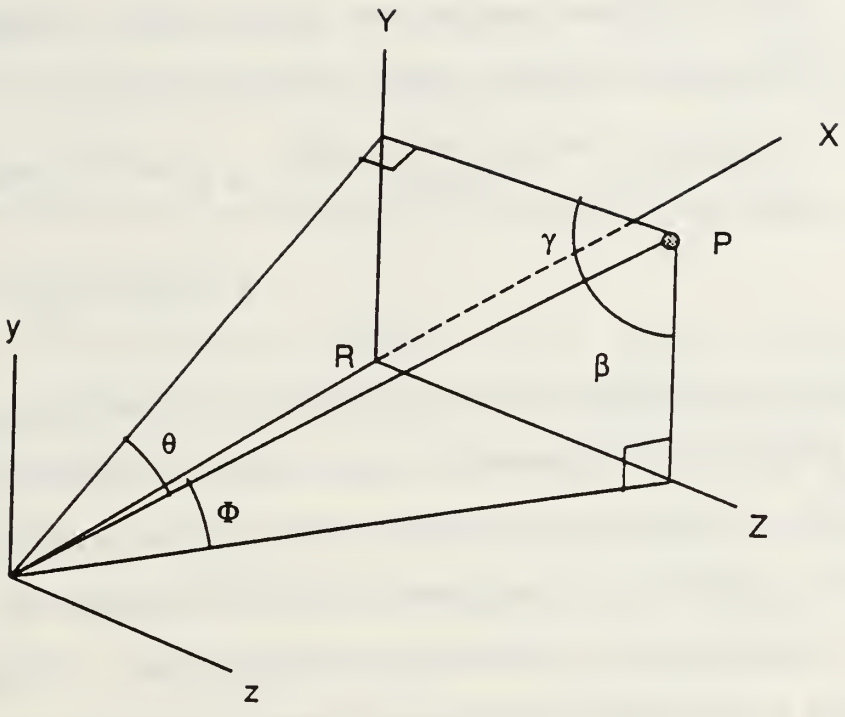

Figure 4 Diffraction Geometry

spatial frequencies k_y and k_z. The third step is simple; the irradiance I is proportional to the time average of $|E|^2$ [Ref. 3], such that

$$I \propto \langle EE^* \rangle \tag{9}$$

B. GENERATION OF WALSH MASK APERTURE FUNCTIONS

The orthogonal Walsh aperture functions were simulated by means of a straightforward algorithm, which is easily described via an analogy with the familiar topic of Fourier transforms. To recover one of the orthogonal trigonometric basis functions upon which Fourier analysis is built, one

could, in principle, calculate the Fourier transform of a Dirac delta function pair [Ref. 5], such that

$$\int_{-\infty}^{\infty} [\delta(f-f_0) + \delta(f+f_0)]e^{-2\pi i f_0 t}\, df = e^{-2\pi i f_0 t} + e^{2\pi i f_0 t}$$

$$= 2\cos(2\pi f_0 t) \qquad (10)$$

While there are more efficient methods of calculating a simple cosine function, the transform approach is conceptually sound. Similarly, if one has at their disposal an efficient Walsh transform routine, it becomes a trivial matter to generate Walsh functions of different sequencies: simply Walsh transform a numerical equivalent of a delta function at the desired sequency.

The mathematical construction of the Walsh masks as outlined above was accomplished by Davis using a fast Walsh transform algorithm which he developed from the generalized transform algorithm theory presented in Elliot and Rao [Ref. 7]. This program was originally written in Microsoft C. Davis translated the code into FORTRAN for incorporation into the mask diffraction program WMASK.for (listed in Appendix A) as a subroutine called FWT. The necessary parameters that govern the mask generation and ultimate diffraction pattern are the desired spatial sequency, the size of the basis set of Walsh functions, and the overall dimensions of the two-dimensional mask matrix which will be produced. For this stage of the project, it was decided that the number of

sequencies in the basis set would remain fixed at 64, corresponding to an eight-by-eight array of square pixels, as shown in Figure 2. The interactive portion of program WMASK.for requests specification of the desired mask sequency in the range zero to 63.

The FWT subroutine operates on a delta function and returns a one-dimensional array of 64 elements corresponding to the discretely sampled Walsh function with the specified sequency number. These 64 elements have assigned values of +1 or -1, representing the transparent and opaque regions, respectively, of the Walsh masks. The program adds the value one to each returned element, and then divides each element by two. This normalizes the sequency array to values of either one (for a transparent pixel) or zero (for an opaque pixel). As a visual check, whose importance was later realized, the program writes the 64 elements to the computer screen in the form of a eight-by-eight matrix. Trial and error simulations and numerical experiments revealed that this basic eight-by-eight matrix format was undersampled for subsequent Fourier transformation [Ref. 5], so the matrix structure was expanded by a factor of two in each dimension. For clarification, the following illustrates the process in generating the Walsh mask for sequency 28 of 63 (the 16-square checkerboard in Figure 2). The 64 element array returned by the subroutine FWT is ordered into an eight-by-eight matrix in the form

```
1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0
0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1
1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0
0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1
```

After the bi-directional expansion by a factor of two, the eight-by-eight matrix becomes a 16 by 16 matrix appearing on screen as

```
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
```

The two-dimensional Fourier transform subroutine used in the program WMASK.for mandates that the input matrix contain complex numbers. The required format calls for the real part of the number followed by the imaginary part, in column order. However, the elements which comprise the Walsh mask are real numbers. For this reason, the numerical model of the Walsh mask as shown above must be modified by injecting a row of zeros, representing the imaginary component of each element,

between each row. This transforms the mask to the proper complex representation. The resulting matrix is 32 by 16.

Finally, the mask was centered on a 64 by 32 zero matrix in complex form, which constitutes a border of zeros, or zeropad, of a factor of two surrounding the mask. This dimension doubling was again reached by trial and error. Its purpose was to interpolate the resulting spectrum via oversampling [Ref. 5]. By increasing the overall dimensions of the input matrix by a factor of two (reaching final dimensions of 32 by 32), the inherent spectral sampling rate is doubled, and consequently, the resulting spatial frequencies in each dimension of the diffraction pattern are then more densely spaced by a factor of two.

As outlined in Press, et al., [Ref. 6], the complex data must first be properly ordered prior to performing the two-dimensional Fourier transform. This ordering is required to reduce the number of steps necessary to complete the transformation. As is typical of many one-dimensional FFT programs, the input data must be reordered in a wrap-around fashion by placing the second half of the data in front of the first half prior to conducting the transform. This quadrant reversal is also discussed in Brigham [Ref. 5]. In two dimensions, with the data in matrix form, this wrap-around procedure must be performed twice, once in each spatial direction. Finally, the matrix is passed to the two-dimensional FFT subroutine in normal FORTRAN order. This

ordering requires partitioning the matrix into a one-dimensional array composed of the sequence of the vertical columns of the input matrix. Such an ordering structure is well represented in Figure 12.11.1 of Press, et al., [Ref. 6]. The FFT subroutine also returns the transformed data in this order, which must be sorted following the reverse of the input procedure, to recover the two-dimensional FFT of the input matrix.

The following illustrates the steps involved in the data manipulation of a simple two-by-two input matrix enroute to the FFT subroutine. The numbers represent the matrix elements. Each step was originally developed as a separate program to verify its function, and then inserted into the master program WMASK.for as subroutines.

1. Input Data Matrix

```
1  2
3  4
```

2. Inject Zeros to Transform to Complex Data

```
1  2
0  0
3  4
0  0
```

3. Center Complex Matrix on Zeropad

```
0  0  0  0
0  0  0  0
0  1  2  0
0  0  0  0
0  3  4  0
0  0  0  0
0  0  0  0
0  0  0  0
```

4. Horizontal Wrap-Around

0	0	0	0
0	0	0	0
2	0	0	1
0	0	0	0
4	0	0	3
0	0	0	0
0	0	0	0
0	0	0	0

5. Vertical Wrap-Around

4	0	0	3
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
2	0	0	1
0	0	0	0

6. Form 1-D Array to Subroutine

4	0	0	3
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
2	0	0	1
0	0	0	0

In returning the transformed data, the steps are taken in reverse.

C. THE FAST FOURIER TRANSFORM

The fast Fourier transform (FFT) has played a significant role in reducing the amount of computing time required for Fourier analysis. In two or more dimensions, as is required by this and other imaging projects, the number of calculations becomes so large that, without the FFT, it could not be done in a useful amount of time. The key feature which makes the FFT work so efficiently is an algorithm to reduce redundant

arithmetic operations. Brigham [Ref. 5] outlines the theory of the FFT and describes its implementation.

The two-dimensional FFT used in this project is an adaptation of one taken from Press, et al., [Ref. 6], which was developed by N. M. Brenner of Lincoln Laboratories. The program provided in the reference was written in FORTRAN. Consequently, all programming for this part of the thesis was done in this language. Although several personal computers were used throughout the programming phase, the majority of the work was performed on a 16-bit Imperial 80286 machine, using the Microsoft FORTRAN 4.1. compiler. To minimize computing time, an executable file of the successfully tested program was installed on an IBM Model 70 PS/2 80386 personal computer with an 80387 coprocessor.

D. CONVERTING THE TRANSFORMED MATRIX INTO DIFFRACTION PATTERNS

The array returned from the FFT subroutine is reordered back into complex matrix form, but does not yet represent the diffraction pattern of the input Walsh mask. In reference to equation (5), the two-dimensional Fourier transform of the aperture function is an expression for the electric field distribution $E(k_y,k_z)$ at the Fourier plane. However, the desired quantity is the irradiance I, as a function of the spatial frequencies.

The algorithm for implementing equation (9) is contained within the subroutine INTENSITY of program WMASK.for. This subroutine converts the returned complex field distribution matrix into a real matrix where each element represents the intensity at a point in the Fourier diffraction pattern. In doing so, the matrix is reduced from dimensions of 64 by 32, to a 32 by 32 square. The matrix then represents the field intensity as a function of spatial frequency, which is the desired form for the diffraction pattern of the Walsh mask aperture function.

A consequence of performing a two-dimensional fast Fourier transform on a square matrix is that, in the final form, the first row and the first column contain all zeros. This result stems from the vertical and horizontal quadrant reversal required in the matrix manipulation prior to transformation. As this first row and column have no real meaning, they are eliminated from the intensity matrix. Therefore, this last operation leaves the matrix in the form of a 31 by 31 square.

E. PLOTTING THE DIFFRACTION PATTERNS

It was necessary to locate a software package which could produce a three-dimensional perspective plot of the intensity matrix as a function of spatial frequency. Such a program is SURFER, which is distributed by Golden Software, Inc. This program was used to produce all of the diffraction pattern plots which appear in Appendix B.

The format required by SURFER demands that the data to be plotted be accessible in three-dimensional Cartesian (x,y,z) coordinates. At this stage in the programming, however, the intensity was represented as a 31 by 31 square matrix. A subroutine called MAT2XYZ was developed to convert the matrix into the required format. In this form, each converted point was assigned a set of (x,y) coordinates to locate the point in the intensity matrix. The (z) coordinate indicated the intensity at that point. Such a process generates 961 data points from the 31 by 31 intensity matrix, and writes them to a disk file for later use by the plotting routine.

SURFER employs a two-step process to create three-dimensional plots of input data. The first step involves forming a regularly spaced grid from irregularly spaced data. The data, in the case of this project, are the (x,y,z) elements which are generated by the master program as described above. These data points are imported from the disk file into SURFER by specification during the gridding phase. In the language of the software, this process is simply called "gridding", and is invoked in the menu heading GRID. The second step in the plotting process calls for loading the gridded data into the subprogram called SURF. SURF is a menu-driven, interactive graphics routine which produces the perspective plots constituting the diffraction patterns. Its output is a plot file which contains the SURFER-generated

surface plot data. This plot file may be viewed on screen or sent directly to a specified plotter.

As noted, SURFER is a menu-driven program. The following parameters were used during the interactive set-up to create the diffraction pattern plots which appear in Appendix B. The parameters are listed in the order in which they were input, and are the result of much trial and error to find the optimal settings. Many of the default parameters were satisfactory for making the diffraction pattern plots; only the parameters which differ from the defaults are listed below.

At the Main Menu:	**GRID**
Filespec to Pass:	**XYZ.GRD**
At Menu:	**Modify**
Select:	**Smooth**
Select:	**Spline** (expansion factor **2**)
Select:	**Begin**

This procedure creates the output grid file OUT.GRD

At the Main Menu:	**SURF**
At Menu:	**Input**
Input grid file:	**OUT.GRD**
At Menu:	**View**
Projection:	**Perspective**
At Menu:	**LineTyp**
Plot constant (x,y,z) :	**X,Y**
At Menu:	**Base**
Plot base:	**No**
At Menu:	**Title**
Plot Title	**Diffraction Pattern For Walsh Mask #28**
Title symbol set:	**DEFAULT.SYM**
Title position:	**(0,-1)**
Title angle in degrees:	**0**
Title character height:	**0.15**
Title color:	**1**
Plot orientation legend:	**No**
Legend position:	**Automatic**

At Menu:	**Axes**
Plot axes:	**X,Y** (each separately)
Axis color:	**1**
Axis symbol set:	**DEFAULT.SYM**
Axis plane:	**Automatic**
Tic distance:	**6.28**
Labeled tic frequency	**2**
Label format:	**Fixed**
Number of decimal digits:	**2**
Label character height:	**0.1**
Tic label angle:	**0**
Axis title:	**Spatial Frequency K Sin A/B**
Title character height:	**0.13**
Title to label distance:	**0.14**

No changes are required in menu headings **Size**, **XYLine**, or **Post**.

At Menu:	**Output**
Name of plot file:	**OUT.PLT**
Scale factor:	**1**
Page position:	**1,1**
Plot file format:	**Binary**
File write mode:	**Overwrite**
Number of decimal digits:	**3**
Send to output device:	**No**

No changes are required in menu heading **Equip**.

This procedure generates the plot file OUT.PLT, which is called for in the final phase of SURFER. This file is sent to the HP7550A plotter.

At Main Menu:	**PLOT**
Name of plot file:	**OUT.PLT**
Do you wish to shift the entire plot?	**No**
Do you wish to scale the entire plot?	**Yes**
Scale factor in X direction:	**0.6**
Scale factor in Y direction:	**0.6**

F. ANALYSIS OF DIFFRACTION PATTERNS

Having established a working diffraction program and stored the listed plotting parameters into SURFER, the remainder of this phase of the thesis research was devoted to generating the Walsh masks and to examining their diffraction pattern plots to reach conclusions about their diffraction behavior. Analysis of the first few diffraction pattern plots immediately raised questions about what they represented; the results did not seem to conform to expectations. The computer code was reviewed to determine where, if any, mistakes were present. "What *should* the diffraction patterns of the Walsh masks should look like?" became the question of interest. Consequently, the project focus shifted to resolve this matter.

G. APPARENT PROBLEMS WITH THE DIFFRACTION PATTERNS

The first few diffraction patterns plotted were very encouraging. In reference again to the Walsh masks in Figure 2, sequency zero appears in the upper left hand corner as the solid black square. When centered on the zeropad, the black square is seen as a square aperture (considering the black portion as transparent, the zeropad as opaque). The diffraction pattern of a square aperture is well known and very identifiable. It appears as Figure 10.25(b) in Hecht [Ref. 3]. It was with much relief that the plot generated by SURFER of Walsh sequency zero (Figure B-1) was identical to

this figure within the scaling limits. Continuing across the first row of masks, with sequencies one, two, etc., the diffraction patterns bore the features which would have been expected for an array of vertical slits. Despite the apparent success, the diffraction pattern produced by sequency 63 (the 64-square checkerboard in Figure 2) did not conform at all to expectations.

This checkerboard pattern has the finest detail of all of the masks in the basis set of Figure 2. Consequently, its diffraction pattern would be expected to be spread out much more than the others. Also, its symmetry about the main diagonal predicts a likewise symmetric diffraction pattern. The plot generated from sequency 63 was neither well dispersed nor symmetric. Only after installing the routine to display the mask on the monitor did the explanation for this become apparent.

1. False Assumptions in the Sequency Ordering

From the outset, it was assumed that the Walsh masks of Figure 2 were sequentially ordered left-to-right in a raster fashion, beginning at the upper left hand corner. Certainly, the black square represents sequency zero. In this system then, the masks of this figure appear "numbered" in sequency as

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

In such a system, the first row appears in order, and the checkerboard is located in position 63.

The revelation caused by the routine to display the masks indicated an entirely different ordering. By calling for the masks in ascending order (from zero), it became evident that instead of a raster structure, the subroutine FWT returned the data in a serpentine fashion. Regardless of this understanding, the first row appears identically in each scheme. This explains why the square aperture pattern could be generated without full comprehension of the sequency ordering. However, the mistakenly assumed raster ordering places the checkerboard in a different location as opposed to where this mask properly appears when located with a serpentine ordering. Therefore, rather than generating the diffraction pattern of the checkerboard by specifying sequency 63, the diffraction pattern of the mask located in position 56 was being produced, which is predictably neither well dispersed nor symmetric.

2. The Proper Ordering of the Walsh Masks

With the preceding discussion in mind, the proper "numbering" of the Walsh masks as generated by the diffraction pattern program and appearing in Figure 2 is

0	1	2	3	4	5	6	7
15	14	13	12	11	10	9	8
16	17	18	19	20	21	22	23
31	30	29	28	27	26	25	24
32	33	34	35	36	37	38	39
47	46	45	44	43	42	41	40
48	49	50	51	52	53	54	55
63	62	61	60	59	58	57	56

Throughout the remainder of this thesis, the Walsh masks and their subsequent diffraction patterns will be referred to using this system, whereby the assigned number indicates both the sequency of the mask and its location in the above table.

H. SYMMETRIES IN THE WALSH MASKS

With a complete understanding of the ordering of the masks, it was quickly determined that the program had been working all along, and production of the diffraction pattern plots resumed in earnest. However, not all of the masks were used. Collectively, an examination of the masks in Figure 2 reveals that masks reflected across the main diagonal are identical except for a 90 degree rotation. This symmetry allows one to draw conclusions about the complete set of masks by examining only the upper (or lower) triangle of the masks in this figure. Such a symmetry reduces the number of required analyses and plots from 64 down to 36, as the

diffraction pattern of a rotated mask is the same as its non-rotated counterpart; the only difference being that the pattern is also rotated by the same amount as the mask. The diffraction patterns which comprise Appendix B, then, are taken from the 36 masks which appear on and above the main diagonal of Figure 2.

I. A TYPICAL DIFFRACTION PATTERN PLOT

Figure 5 shows the diffraction pattern for Walsh mask 28. This mask is the 16-square checkerboard. This typical pattern illustrates the type of information which is analyzed when examining these plots. Angles A and B correspond to angles ϕ and θ shown in Figure 4; they relate to, respectively, the angular spread due to diffraction in the Z and Y directions of the Fourier plane. The axes are labeled in normalized units of spatial frequency corresponding to k_y and k_z. Each direction is labeled out to values of $\pm 2\pi$. The multiples of π are used to keep the dimensions consistent in terms of spatial frequency. These units of spatial frequency are the reciprocal of the Walsh mask pixel size.

The irradiance (vertical) axis is neither labeled nor scaled. This fact is due to the nature of the Walsh masks. All masks (except for the square aperture of sequency zero) have equal transparent and opaque areas. Therefore, the diffraction patterns of all of the masks will transmit the same amount of electromagnetic energy, albeit in different

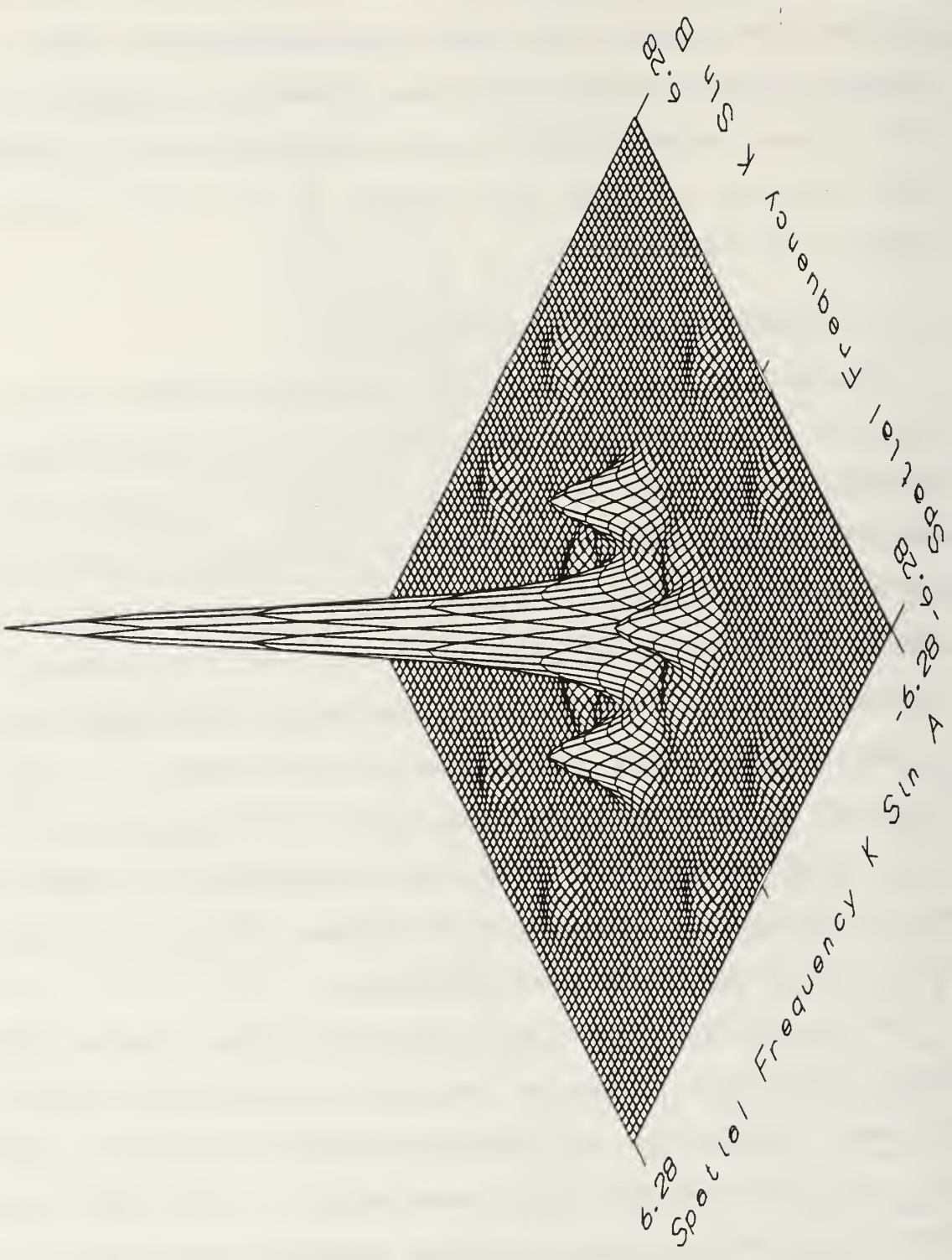

Figure 5 Diffraction Pattern for Sequency 28

distributions. Hence, the relative irradiance scaling is the same for all masks, and absolute scaling is unnecessary.

J. CONCLUSIONS CONCERNING THE DIFFRACTION LIMIT

The quantity which concerns us is the spreading of the irradiance in each diffraction pattern. Although all of the patterns show a characteristic central peak containing the majority of the irradiance, some display a significant amount of energy spilling out into the periphery of the pattern. For reference purposes, Appendix B constitutes a catalog of all 36 different patterns generated by the Walsh masks in the basis set of 64. They may be examined individually to determine the limit in terms of the spatial frequency necessary to retain the diffracted irradiance; or, perhaps more expeditiously, only the extreme cases need be considered. The result of such a process constitutes the answer to the research question at hand.

In reference to the Walsh masks in Figure 2, the finest detail occurs in mask 56, the dense checkerboard pattern. Certainly its diffraction pattern must be considered as a limiting case. Nearly all of the masks in the first row show diffraction patterns with wide angular dispersion as well, particularly masks 1, 2, and 3. Such patterns define the extremes of diffraction spreading for this basis set of Walsh masks. It is important to ensure that the limit chosen encloses nearly all of the significant irradiance of even the

most severe case of diffraction. After careful consideration, it was determined that this worst case occurs for Walsh sequency three (Figure B-4).

K. FORMULATING THE DIFFRACTION LIMIT

Having made the above determination, it became a matter of simply reading from the plot the minimum spatial frequency required to encompass the majority of the diffracted irradiance. It was concluded that this minimum spatial frequency was 6 reciprocal length units, as defined above. To put this result in a form useful in future instrumental design, it may be stated that the following relation must be satisfied

$$\left|\frac{2\pi}{\lambda} \sin\theta\right| \leq 6 \ . \tag{11}$$

The angle θ is used here since, once a mask is "rotated", in that its counterpart across the main diagonal is used, the angle A becomes equivalent to the angle B. For simplicity, the diffraction angle henceforth shall be called θ. Figure 6 illustrates the simplified relationship between the mask, the optics, and the detector.

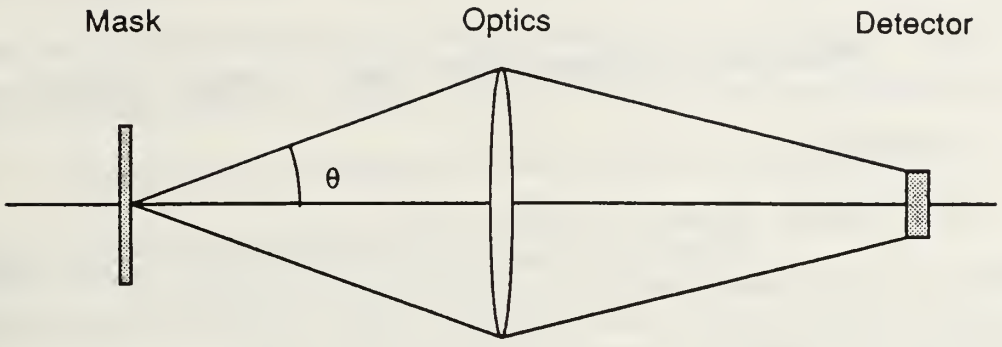

Figure 6 Mask, Optics, and Detector Orientation

The formulation of equation (11) constitutes the solution to the first thesis problem.

IV. RESOLUTION CRITERIA FOR IMAGE PATTERN RECOGNITION

A. OVERVIEW

The second phase of this research project involved, by means of computer simulation, a determination of what minimal spatial resolutions are needed for pattern recognition in images of commonplace objects. Establishment of these criteria are of great importance in the overall multiplexed imaging research project, for they dictate the minimal number of picture elements (pixels) needed by a practical remote sensing instrument. This minimal number of pixels, in turn, specifies the number of Walsh basis functions and encoding masks required to accomplish this task.

A working prototype system does not yet exist; therefore this problem was addressed in reverse. Digitized images were transformed into a sets of sequency spectra. These spectra were sequency-truncated (low-pass filtered) to mimic the effects of limited resolution in an actual instrument. The filtered sequency spectra were then re-transformed into the image domain for analysis.

The procedure may be further clarified by recalling that an image's higher spatial frequencies are responsible for finer image detail. In this research, spatial sequencies play the role normally reserved for spatial frequencies with regard

to traditional image analysis approaches. Elimination of higher sequencies degrades image resolution in the same sense that the fine detail in an observed field is degraded by the loss of components of higher-order spatial frequency.

B. THE WALSH FUNCTIONS AND IMAGE PROCESSING

Previous sections have discussed the Walsh functions with regard to sequency theory and the proposed encoding technique. Beauchamp [Ref. 4] goes much further in detailing several other applications of the Walsh functions in image processing, including pattern recognition, and image enhancement, transmission, and restoration. However, those properties of the Walsh functions which are summarized in Chapter II of this thesis are sufficient for a description of the procedures used in this phase of the research.

C. CREATING DIGITIZED IMAGES

The actual order of tasks for determining the number of sequencies required for addressing the image resolution - pattern recognition problem were as follows:

- Capture a series of images on a videocamera
- Digitize the images into computer disk files
- Write software to store, access, and display the images
- Write software to perform the sequency low-pass filtering
- Analyze the resulting images to establish appropriate resolution criteria.

D. CAPTURING THE IMAGES

The original plan was to record still images and to perform a digitizing scan of each using a LYNXX solid-state CCD slow-scanning imaging camera. Prof. D.L. Walters procured such a camera system along with the necessary imaging software, from SpectraSource Instruments. However, the camera system was modified for higher frame rates, and was shipped back to the manufacturer. Hence, because the system was not available for use in this project, an alternative was sought.

The most readily available device for image recording is an ordinary VHS camcorder. A Panasonic VHS-C Palmcorder was borrowed from a friend to record the images used in this project. A series of 15 images were recorded from areas of interest around the Monterey Peninsula, with the understanding that the VHS tape would be played back in a freeze-frame mode into digitizing equipment operated by the NPS Physics Department. The output from this equipment would then be a digitized disk file of each image in simple binary format.

E. DIGITIZING THE IMAGES FROM VHS TAPE

Difficulties arose with regard to gaining access to the original image digitization equipment, due to the author's lack of familiarization with it. Therefore, a quest to locate alternate means to complete the task was undertaken. An ideal system would grab a frame from the VHS playback, scan it, digitize it, and then write the image to an binary disk file.

Several systems were located throughout the NPS campus which apparently had this capability.

1. Apple Macintosh IIx Programs

Two programs in use at the NPS Linear Accelerator facility were offered by Prof. X.K. Maruyama, and each was thoroughly investigated for suitability to the image digitization task. The programs are IMAGE 1.29, and PIXELWORKS. Both programs run on an Apple Macintosh IIx personal computer, and have similar capabilities well beyond the simple needs of this project. The programs are able to manipulate and digitize an image, which is played back from the camcorder into the video-in port of the Macintosh. Difficulties arose at this stage of the process.

The programs offer a wide range of digitized data formats; in particular, TIFF (and simple TIFF), PICT, MacPaint, Palette, and RAW, are available. However; binary is not. Substantial effort was made to resolve this problem, but despite a collaborative effort, no simple solution was found. As an alternative approach, Davis suggested that if the format of the data type was known, he could then write a program to convert the data into binary bytes. Unfortunately, the supporting documentation for the Macintosh imaging software was not detailed enough to provide this information. Therefore, further investigation into this translation method were halted, and alternate digitization means were sought.

2. Other Means of Digitization

Earlier failures in producing a digitized image disk file in binary format lead to an expanded search for an alternate means to accomplish this task. Several Hewlett-Packard page scanners are located in different departments at NPS, and were considered as possible candidates. The NPS Computer Center also uses HP scanners, as well as an IBM page scanning system. The IBM system appeared most promising, but it proved to demand a critical investment in terms of training time and familiarization. Furthermore, it was perpetually oversubscribed. Eventually, these page scanners were dropped as alternate means of digitization.

3. Successful Digitization

Davis contacted Prof. A.W. Cooper and Mr. W.J. Lentz of the NPS Physics Department, who agreed to loan some of their equipment and expertise to assist the author with the image digitization. Their equipment configuration mandated that the original VHS-C camcorder images be translated onto a standard VHS cassette tape. This was easily done using the playback feature of the camcorder in conjunction with the record mode of the author's VHS recorder.

The digitizing equipment made use of integrated software coupled through an IBM personal computer to link a playback VCR and a display monitor. The digitizing program was DT - IRIStutor V01.01, which is marketed by DT, Inc. This program

utilizes an IRIStutor DT2861 frame-grabber board in conjunction with the IBM PC to reduce a captured image to a 512 by 512 pixel digitized image file, in eight bit binary format. The images from the VHS tape were subsequently digitized and stored on 5-1/4 inch 360k byte diskettes, with one image per diskette.

F. CONVERTING AND DISPLAYING THE IMAGES ON SCREEN

The files generated by DT-IRIS contained a header block of 512 bytes prior to the actual pixel-by-pixel digitized representation of the image. It was necessary to strip off this information prior to displaying the image on the VGA monitor of the project's IBM PS/2 model 70 PC. In order to accomplish this, Davis wrote a program in Microsoft C5.1 called C512, which is used to convert the 512 by 512 DT-IRIS images into simple eight bit binary format. This program is included as Appendix J.

Once the image file had been converted into usable form, it was necessary to write another routine to display the image on the monitor. Again programming in C, Davis developed the program D512, which accesses the binary image file, and displays it in 16-level greyscale on the VGA monitor. In this scale, a shade of zero represents black, while a shade of 15 is bright white. To accommodate the line restrictions imposed by the monitor, the program reduces the display to 480 rows by 512 columns. This is accomplished by stripping off the top

16, and the bottom 16, horizontal lines of the image. Thus, in the absence of any sequency filtering, the program displays the image in 480 by 512 pixel format. The program D512 is included as Appendix K.

G. FIVE REPRESENTATIVE IMAGES

Originally, a total of 15 scenes were recorded using the Panasonic camcorder. These 15 scenes were viewed with Davis to select those which contained characteristic features and/or desirable symmetries which might prove interesting in the context of the low-pass sequency filtering problem. The following images were chosen:

- U.S. Coast Guard Cutter Point Barrow
- U.S. Coast Guard Lighthouse at Point Pinos
- A friendly German shepherd
- A house with a picket fence
- U.S. Coast Guard emblem

Despite the apparent Coast Guard bias in subject material, the images selected represent a broad range of spatial characteristics such as fine detail, periodic structure, sharp contrast, circular symmetry, and so forth. The original, unprocessed, images appear in the Figures 7 through 11. Note that each figure is captioned with a noun descriptor followed by a number. This naming convention has been adopted

throughout the remainder of the research. The noun name simply describes the dominant object in the image; the number indicates the number of sequencies used in each direction in the image basis set. Using this scheme, the first image is Cutter512, indicating that it is the Coast Guard cutter with no sequencies eliminated (i.e., 512 by 512 pixels).

This labeling system is used throughout the project and is most useful when studying the images contained within Appendices C - I. However, there is one departure from the system. Images labeled with a "V" or an "H" postscript indicate that the image was generated using only a one-dimensional Walsh transform, with that transform applied in the vertical (V) or horizontal (H) direction. These descriptors appear only in conjunction with the sequency filtering applied to the image of the cutter, and are contained in Appendices C and D.

All of the images which appear in the appendices were photographed directly off of the VGA monitor. For this reason, the curvature of the screen is noticeable. The small black tab which seems to move about on the left and lower borders is a marker placed by the author to identify the developed prints. The shaded diagonal strip which is evident in some of the photographs, such as in Cutter2V in Figure C-9, is a consequence of the monitor's 1/30 second refresh and flyback time when photographed. The pictures were taken with black and white ISO 400 KODAK Tri-X pan film, at an aperture

setting of f4, and shutter speed of 1/15 second. The camera was a Canon AE-1.

H. PROCEDURES BY WHICH IMAGE RESOLUTION WAS MODIFIED

The FORTRAN program IMAGE.for, written by the author, is the primary program which simulates image resolution degradation. It does this by taking an original, full resolution, digitized image produced by program C512, and operates on it to generate a new file with degraded resolution. The degradation involves three steps. First, the program calculates the Walsh transform of the full-resolution image, to produce its sequency spectrum. Second, it performs low-pass sequency filtering on the image spectrum, with the maximum sequency cutoff specified by the user. Third, it calculates the inverse Walsh transform of the filtered spectrum, yielding a modified image. Each of these steps is described in detail below. The reader may also wish to consult Appendix L, which contains a listing of program IMAGE.for.

1. Production of the Image's Sequency Spectrum

The program IMAGE.for uses the same fast Walsh transform subroutine (see subroutine FWT in Appendix L) as was incorporated into program WMASK.for, described earlier in section III.B. of this thesis. Two different types of transform geometries were used in this research. For most of the images of interest, a straightforward two-dimensional

Figure 7 Cutter512

Figure 8 Lhouse512

Figure 9 Dog512

Figure 10 Fence512

Figure 11 Emblem512

resolution degradation was required, with the same sequency filter cutoff in both X and Y directions. A second approach, which was applied only to the Cutter image, investigated the effects of one-dimensional sequency filtering along both the X and Y directions, independently, while leaving the other directions unfiltered.

The one-dimensional transform procedure is the simpler of the two. If resolution degradation is desired in the X (horizontal) direction, the image is first partioned into rows. The program calls the FWT subroutine for that row, and returns its sequency spectrum. It then cycles to the next row, and loops until all rows' spectra have been computed. The one-dimensional transform for resolution degradation in the Y (vertical) direction is effectively the same, except that the program operates on columns, rather than rows.

The two-dimensional transform operation is essentially a combination of both one-dimensional versions. The program first performs the Walsh transform on each column of the image in turn, then it calculates the transform for each row. This yields the two-dimensional sequency spectrum of the original spectrum. Clearly, the efficient implementation of such a numerically intensive calculation is completely dependent upon a fast transform algorithm; direct calculation would be prohibitively slow. Even with the FWT, full two-dimensional Walsh spectrum calculations for a 512 by 512 pixel image

required nearly 300 seconds on a 25 MHz 80386 PC, using an 80387 math coprocessor.

2. The Sequency Filtering Operation

When the program IMAGE.for is first invoked, it prompts the user for sequency filter cutoff information. The actual sequency filtering operation is then quite simple. The program cycles through each sequency spectrum row and/or column that is to be filtered, and sets spectral samples at sequencies beyond the cutoff sequency to zero. This is not unlike the zeropad procedure described in section III.B., although in this case the size of the spectrum array is not increased.

3. Calculation of the Degraded Image

The inversion of the filtered sequency spectrum to its corresponding image is straightforward: IMAGE.for calls FWT to calculate the inverse Walsh transform of each row and/or column of the spectrum to yield the image. Fortunately, this is a particularly simple procedure using the type of algorithm developed by Davis for subroutine FWT. When the data are retained in increasing sequency order (as they are here), and the binary bit-reversal sorting portion of the fast transform is followed by a decimation in sequency, the Walsh transform and its inverse are algorithmically identical, except for a simple multiplicative scaling factor following the inverse

transform operation. This property greatly simplified the writing of program IMAGE.for

I. ANALYSIS OF RESULTS OF THE RESOLUTION DEGRADATION PROCESS

A complete set of resolution degradation results for each of the five original images is included in its own appendix. The reason for this is that the conclusions which are drawn from these series of image photographs are highly subjective; it is not expected that all readers of this thesis will agree completely with the interpretations. Each series of sequency filtered degradation is represented in it's entirety for the reader to confirm or refute the author's analysis. For illustrative purposes, the image of the Coast Guard cutter was degraded in three separate manners; once vertically, once horizontally, and once in two dimensions. All other images were degraded in two dimensions only. The appendices contain the following series of images and should be consulted throughout the following discussion.

- Appendix C - Cutter(512 - 0)V
- Appendix D - Cutter(512 - 0)H
- Appendix E - Cutter(512 - 0)
- Appendix F - Lhouse(512 - 0)
- Appendix G - Dog(512 - 0)
- Appendix H - Fence(512 - 0)
- Appendix I - Emblem(512 - 0)

1. One-Dimensional Sequency Filtering

a. Vertical Filtering of the Cutter Image

Referring to the images of the cutter in Appendix C, the effects of vertical sequency filtering are clearly illustrated. Figure C-1 is the unfiltered image of the cutter, accordingly titled Cutter512. In each successive figure, the vertical image resolution is made coarser by a factor of two. For example, Figure C-2 represents Cutter256V, and is produced from half as many sequencies as Cutter512. Figure C-3 represents Cutter128V, which is produced from half as many sequencies as Cutter256V, and so forth. This pattern of resolution reduction (in both one and two dimensions) continues throughout this and all subsequent series of image degradations.

The basic structure of Cutter512 in Figure C-1 indicates four vertically spaced regions corresponding to the sky, the hillside, the cutter and its moorings, and the water in the foreground. With such an orientation of dominant features, one would expect that vertical filtering would preserve these regions until only the lowest vertical sequency components remained. This, in fact, is the case. In Figures C-1 through C-5, corresponding to vertical resolutions of 512, 256, 128, 64, and 32 pixels, this four-region structure remains obvious. It begins to become unidentifiable only at

vertical resolutions of 16 or less, whereas the cutter itself loses recognizable form at a sequency cutoff of 64.

A noticeable feature of this and the following image degradation series is the apparent white "spots" which seem to outline the edges of the image. Cutter256 in Figure C-2 exhibits this behavior quite clearly. It is hypothesized that these spots are the result of the type of "ringing", or overshoot phenomenon, associated with the processing of adjacent areas of high contrast. If this is the correct explanation for their presence, then there are more than likely just as many black "spots"; the eye just does not see them standing out as clearly. The nature of these spots and their effect on the imaging process are offered as recommended areas of future research.

b. Horizontal Filtering of the Cutter Image

Appendix D contains the horizontally degraded series of images of the cutter. As before, the first image in the series, Cutter512 in Figure D-1, represents the image in its undegraded form. All subsequent images in this series are degraded by a factor of two in the horizontal direction.

Cutter512 displays much greater spatial variation in the horizontal as compared to the four dominant vertical regions discussed above. Therefore, one would expect that eliminating the higher order sequencies in the horizontal would have less effect on the overall image degradation. This

expectation is validated by comparing Cutter256H in Figure D-2 with the original, undegraded image. When making this same comparison with Cutter256V (Figure C-2), it becomes readily apparent that horizontal filtering detracts less from the image quality (as opposed to vertical filtering) for this particular image, due to it's spatial characteristics.

Another easily seen result of horizontal filtering of the cutter image is the rapid degradation of the narrow, vertical features. The cutter's mast, for example, may be described as such, and is produced from higher order sequencies. One would then expect that the mast would quickly lose its detail while the image underwent horizontal sequency filtering. This is indeed the case. The mast becomes largely indistinguishable in Cutter128H (Figure D-3), while the overall image of the ship is still recognizable in Cutter32H (Figure D-5).

2. Two-Dimensional Sequency Filtering

a. The Coast Guard Cutter

Here, and for all subsequent image series, resolution is diminished by a factor of two in both vertical and horizontal directions for each successive degradation. The two-dimensional filtering of this image is displayed in Appendix E, and is essentially the combination of the one-dimensional horizontal filtering as described above and the one-dimensional vertical filtering. Analysis of this series

concludes that although the four distinct vertical regions (the sky, hillside, moorings, and water) are still visible beyond a sequency cutoff of 64, the spatial structure of the cutter itself in Figure E-4 is no longer recognizable.

b. The Lighthouse

The series of degradations of this image appear in Appendix F. The undegraded image, Lhouse512 in Figure F-1, seems partitioned into two dominant vertical regions, and three horizontal. However, none of these regions incorporate any fine detail in either dimension. Therefore, one would expect that the image pattern would remain recognizable even with comparatively few low-order sequencies. Lhouse64 in Figure F-4 confirms this, as the lighthouse is clearly recognizable when formed from 64 resolution elements. In fact, Lhouse32 in Figure F-5, is still largely identifiable from only half again as many elements.

c. The German Shepherd

This series of degradations appears in Appendix G. The undegraded image in Figure G-1 indicates that there are few, if any, right angles in the scene. This figure also shows a great amount of fine detail, as well as a somewhat circular symmetry around the dog's head. Therefore, one would expect that the pattern would require most of the higher order sequencies to remain identifiable. This is not so ! Facial characteristics, such as eyes, nose, and mouth, appear

discernible even with resolution degraded to 32 sequencies as seen in Dog32 in Figure G-5. There is most likely a strong psychological reason for this. Perhaps the visual information processing areas of the human brain selectively register this pattern as "wolf!" or "danger!", even with very limited spatial image information.

d. The House with the Picket Fence

This image was chosen for analysis because of the pronounced periodic structure exhibited by both the fence and the support columns on the front porch. This is evident from Figure H-1. Both features are vertical structures of high contrast, and should be susceptible to modification by horizontal sequency filtering. The spatial period of the columns is greater than that of the fence. Therefore, one would predict that the fence detail would be filtered out first. Fence256 and Fence128, in Figures H-2 and H-3, respectively, bear this prediction out. Nonetheless, the object of primary concern with regard to pattern recognition is the house itself, and it loses recognizable form beyond resolution with 64 elements.

e. The Coast Guard Emblem

This image was selected for analysis because it represents several types of symmetry whose degradation effects are difficult to predict. The undegraded image in Figure I-1 shows some horizontally varying structure in the bars of the

shield, but the overall symmetry is circular. Otherwise, there is very little purely horizontal or purely vertical detail, so that one is left to empirical judgment to evaluate results.

Emblem64 in Figure I-4 contributes to the conclusion that the shield structure becomes unrecognizable beyond a sequency cutoff of 64. However, the bulk circular outline of the anchors and lifering are still apparent; in fact, they remain quite visible in Emblem32 in Figure I-5. This is somewhat surprising because these high contrast, high sequency features do not suffer rapid image degradation while undergoing sequency filtering. As was the case with the image of the dog, results did not agree closely with predictions. The common feature between these two departures from prediction is the circular symmetry. Recall that the Walsh functions which were used throughout this project were generated in rectangular coordinates. Perhaps there is a connection to the coordinate system of the applied Walsh transform and the dominant symmetry of the observed scene. Further research into this apparent consequence is in order.

J. DETERMINING THE PATTERN RECOGNITION RESOLUTION LIMIT

From the sequency filtered degradations of the five images, the author concludes that, with the exception of the Emblem series, the images require *at least* 64 sequencies for minimum pattern recognition. Again, it must emphasized that

this is a somewhat subjective judgment. The Emblem series is excluded from this general statement because, as the reader should concur, it remains distinguishable beyond this sequency cutoff, down to a minimum of 32. It is believed that this is related to the highly circular nature of this particular image.

Nonetheless, it is with confidence that this phase of the research concludes that the minimum number of sequencies required for threshold pattern recognition is 64. This result allows one to quantify several physical parameters. For example, this sequency limit mandates that 64 sequencies be available to form the basis set of orthogonal functions in each of the X and Y spatial directions. Furthermore, the number of pixels required to display the output image must be fixed at 4096, which represents the square of 64.

V. CONCLUSIONS AND RECOMMENDATIONS

The intent of the initial task of this project was to mathematically model the two-dimensional sequency-ordered Walsh functions, and to determine the diffraction pattern of each. This was accomplished. The result of studying the patterns lead to the formulation of the following equation which quantifies the diffraction limited performance of the encoding scheme:

$$\left|\frac{2\pi}{\lambda} \sin\theta\right| \leq 6 \ .$$

The angle θ represents the maximum tolerable diffraction angle which will ensure that all of the diffracted photon flux passing through the Walsh mask will be focused onto the detector. This angle, coupled with the size of the detector and the wavelength of operation, provides useful information to the follow-on work which will proceed to design the optics for a prototype instrument. This simple relationship is illustrated in Figure 6.

The second question addressed by this thesis was directed towards establishing the minimum number of sequencies required to produce a recognizable image. This was accomplished by generating images with increasing numbers of higher order sequencies filtered out. Once the image pattern could no

longer be deemed identifiable, it was concluded that the minimum pattern recognition limit had been reached. From examining the all of the series of sequency filtered images in Appendices E - I, it was concluded that no less than 64 sequencies are required to produce a recognizable pattern.

It is recommended that future thesis work be directed toward investigating how the coordinate system of the Walsh functions affect the results. This question is raised from analysis of the series of degraded images of the circular Coast Guard emblem (Appendix I). It appears that perhaps the dominant symmetry of the object plays an important role in determining the minimum number of sequencies required to resolve it. An interesting project may be to regenerate the findings of this project while using Walsh functions derived under a different coordinate system, and make comparisons between the two.

Furthermore, queries into the dependency of image resolution as a function of the available greyscale level, and the cause and effects of the "ringing" phenomenon apparently caused by processing adjacent elements of high contrast, are recommended. Either aspect could possibly affect the findings of this thesis, and may mislead future research based upon the conclusions drawn herein.

APPENDIX A

```
*     LT B .J. MUSSELMAN
*
*     THESIS PROGRAM  -  WMASK.FOR
*
*     THIS PROGRAM COMPUTES MATHEMATICAL MODELS OF WALSH SEQUENCY
*     MASKS, AND CALCULATES THEIR DIFFRACTION PATTERNS THROUGH
*     2 - D FOURIER TRANSFORMS.  THE COMPLEX TRANSFORM MATRIX IS
*     REDUCED TO AN INTENSITY MATRIX, WHICH IS STORED IN CARTESIAN
*     COORDINATES (X,Y,Z) IN A FILE CALLED C:\FORTRAN\TEMP\XYZ.DAT
*
*     THE INPUT MATRIX (WALSH MASK) IS 16 X 16, AND IS CENTERED
*     IN A  32 X 32 ZEROPAD.  THE ENTIRE MATRIX IS SENT TO THE
*     2 - D FOURIER TRANSFORM SUBROUTINE.
*
      INTEGER ISIGN,NDIM,ROW,NROW,COL,NCOL,NN,N,R,J,SEQ,NUMSAMPLS
      DIMENSION NN (2)
      REAL IDATA(64,32),TDATA(2048),PDATA(1024)
      REAL W(64),X(961),Y(961),Z(961),TEMP,EESTAR(32,32)
      REAL A(8,8),B(8,16),C(16,16),CLIN(256)
      NDIM = 2
*
*     NN (1) REPRESENTS ROWS, NN (2) REPRESENTS COLUMNS
*
      NN(1) = 32
      NN(2) = 32
*
      NCOL = 32
      NROW = 32
*
      ISIGN = 1
      NUMSAMPLS = 64
*
      OPEN (11,FILE = 'C:\FORTRAN\TEMP\XYZ.DAT', STATUS = 'OLD')
*
      WRITE(*,*) 'WMASK.FOR IS CURRENTLY CONFIGURED TO PRODUCE A BASIS'
      WRITE(*,*) 'SET OF 64 WALSH SEQUENCIES NUMBERED FROM 0 TO 63.'
      WRITE(*,*)
      WRITE(*,*) 'ENTER SEQUENCY NUMBER TO GENERATE AND TRANSFORM :'
      READ(*,*) SEQ
*
      DO 52 ROW = 1, NROW * 2
         DO 52 COL = 1, NCOL
            IDATA(ROW,COL) = 0.0
   52 CONTINUE
*
*
      DO 10 J = 1, NUMSAMPLS
   10    W(J) = 0.0
      W(SEQ + 1) = 1.0
*
      CALL FWT (NUMSAMPLS,W,.FALSE.)
*
*     THIS SECTION DEVELOPS THE 2X EXPANDED MATRIX USING WALSH
*     SEQUENCIES AS RETURNED FROM SUBROUTINE FWT.  THE MATRIX
*     IS WRITTEN TO THE SCREEN AS A VISUAL CHECK.
*
      J = 1
      DO 32 ROW = 1, SQRT(NUMSAMPLS)
         DO 32 COL = 1, SQRT(NUMSAMPLS)
```

```
            A(ROW,COL) = .5 * (W(J) + 1)
            J = J + 1
   32 CONTINUE
      DO 43 ROW = 1, SQRT(NUMSAMPLS)
         DO 43 COL = 1, SQRT(NUMSAMPLS)
            B(ROW, 2 * COL) = A(ROW,COL)
            B(ROW, 2 * COL - 1) = B(ROW, 2 * COL)
   43 CONTINUE
      DO 34 ROW = 1, SQRT(NUMSAMPLS)
         DO 34 COL = 1, 2 * SQRT(NUMSAMPLS)
            C(2 * ROW , COL) = B(ROW,COL)
            C(2 * ROW - 1,COL) = C(2 * ROW,COL)
   34 CONTINUE
      DO 35 ROW = 1, 2 * SQRT(NUMSAMPLS)
         WRITE(*,201) (C(ROW,COL) , COL = 1, 2 * SQRT(NUMSAMPLS))
   35 CONTINUE
      READ(*,*)
*
      J = 1
      DO 36 ROW = 1, 2 * SQRT(NUMSAMPLS)
         DO 36 COL = 1, 2 * SQRT(NUMSAMPLS)
            CLIN(J) = C(ROW,COL)
            J = J + 1
   36 CONTINUE
*
*
*     FORM THE COMPLEX MATRIX WITH ALL IMAGINARY COMPONENTS ZERO
*
      CALL ZEROPAD (CLIN,IDATA)
*
      CALL FLIPFLOP(IDATA,NROW,NCOL)
*
      CALL MAT2LIN (NROW,NCOL,IDATA,TDATA)
*
      CALL FOURN (TDATA,NN,NDIM,ISIGN)
*
      CALL LIN2MAT (NROW,NCOL,IDATA,TDATA)
*
      CALL FLIPFLOP (IDATA,NROW,NCOL)
*
      CALL INTENSITY (NROW,NCOL,IDATA,EESTAR)
*
      CALL MAT2XYZ (NROW,NCOL,EESTAR,X,Y,Z)
*
*     WRITE XYZ MATRIX TO FILE FOR PLOTTING
*
      DO 15 N = 1, (NROW - 1) * (NCOL - 1)
         WRITE(11,102) X(N), Y(N), Z(N)
   15 CONTINUE
*
  100 FORMAT(1X,F5.2)
  101 FORMAT(1X,8(F8.2,2X))
  102 FORMAT(1X,F8.3,2X,F8.3,2X,E12.6)
  201 FORMAT(1X,16(F4.1,1X))
      END
*
      SUBROUTINE ZEROPAD (CLIN,IDATA)
*
      INTEGER J,ROW,COL
      REAL CLIN(1024),IDATA(64,32)
```

```
*
      J = 1
      DO 12 ROW = 17,47,2
         DO 12 COL = 9,24
            IDATA(ROW,COL) = CLIN(J)
            J = J + 1
   12 CONTINUE
      RETURN
      END
*
      SUBROUTINE MAT2LIN (NROW,NCOL,IDATA,TDATA)
*
      INTEGER N,ROW,COL
      REAL IDATA(64,32),TDATA(2048)
*
      N = 1
      DO 4 COL = 1, NCOL
         DO 4 ROW = 1, NROW * 2
            TDATA(N) = IDATA(ROW,COL)
            N = N +1
    4 CONTINUE
      RETURN
      END
*
      SUBROUTINE LIN2MAT (NROW,NCOL,IDATA,TDATA)
*
      INTEGER N,COL,ROW
      REAL IDATA(64,32),TDATA(2048)
*
      N = 1
      DO 6 COL = 1, NCOL
         DO 6 ROW = 1, NROW * 2
         IDATA(ROW,COL) = TDATA(N)
         N = N + 1
    6 CONTINUE
      RETURN
      END
*
      SUBROUTINE INTENSITY (NROW,NCOL,IDATA,EESTAR)
*
      INTEGER N,ROW,COL
      REAL IDATA(64,32),PDATA(1024),EESTAR(32,32)
*
      N = 1
      DO 8 COL = 1, NCOL
         DO 8 ROW = 1, (2 * NROW) -1, 2
            PDATA(N) = (IDATA(ROW,COL))**2 + (IDATA(ROW + 1,COL))**2
            N = N + 1
    8 CONTINUE
*
      N = 1
      DO 9 COL = 1, NCOL
         DO 9 ROW = 1, NROW
            EESTAR(ROW,COL) = PDATA(N)
            N = N + 1
    9 CONTINUE
      RETURN
      END
*
      SUBROUTINE MAT2XYZ (NROW,NCOL,EESTAR,X,Y,Z)
```

```
*
      INTEGER N,ROW,COL
      REAL EESTAR(32,32),X(961),Y(961),Z(961)
*
      N = 1
      DO 2 ROW = 1, NROW - 1
         DO 2 COL = 1, NCOL - 1
            X(N) = -6.2831 + (ROW - 1) * (12.5662/30)
            Y(N) = -6.2831 + (COL - 1) * (12.5662/30)
            Z(N) = EESTAR(ROW + 1,COL + 1)
            N = N + 1
    2 CONTINUE
      RETURN
      END
*
      SUBROUTINE FLIPFLOP (BOX,NROW,NCOL)
*
      INTEGER R,C,NROW,NCOL
      REAL BOX(64,32)
*
*     RE-ORDER MATRIX LEFT TO RIGHT
*
      DO 2 C = 1, NCOL/2
         DO 2 R = 1, NROW * 2
            TEMP = BOX(R,C)
            BOX(R,C) = BOX(R,NCOL/2 + C)
            BOX(R,NCOL/2 + C) = TEMP
    2 CONTINUE
*
*     RE - ORDER MATRIX TOP TO BOTTOM
*
      DO 3 C = 1, NCOL
         DO 3 R = 1, (NROW*2)/2
            TEMP = BOX(R,C)
            BOX(R,C) = BOX((NROW*2)/2 + R,C)
            BOX((NROW*2)/2 + R,C) = TEMP
    3 CONTINUE
      RETURN
      END
*
      SUBROUTINE FOURN (DATA,NN,NDIM,ISIGN)
*
      REAL  WR,WI,WPR,WPI,WTEMP,THETA
      DIMENSION NN(NDIM),DATA(*)
      NTOT = 1
*
      DO 11 IDIM = 1, NDIM
         NTOT = NTOT * NN(IDIM)
   11 CONTINUE
      NPREV = 1
      DO 18 IDIM = 1, NDIM
         N = NN(IDIM)
         NREM = NTOT / (N * NPREV)
         IP1 = 2 * NPREV
         IP2 = IP1 * N
         IP3 = IP2 * NREM
         I2REV = 1
              DO 14 I2 = 1,IP2,IP1
                   IF (I2 .LT. I2REV) THEN
                       DO 13 I1 = I2, I2 + IP1 - 2, 2
```

```
                              DO 12 I3 = I1, IP3, IP2
                                   I3REV = I2REV + I3 - I2
                                   TEMPR = DATA (I3)
                                   TEMPI = DATA (I3 + 1)
                                   DATA (I3) = DATA (I3REV)
                                   DATA (I3 + 1) = DATA (I3REV + 1)
                                   DATA (I3REV) = TEMPR
                                   DATA (I3REV + 1) = TEMPI
   12                         CONTINUE
   13                    CONTINUE
                      ENDIF
                      IBIT = IP2/2
    1                 IF ((IBIT .GE. IP1) .AND. (I2REV .GT. IBIT)) THEN
                         I2REV = I2REV - IBIT
                         IBIT = IBIT/2
                         GO TO 1
                      ENDIF
                      I2REV = I2REV + IBIT
   14            CONTINUE
                 IFP1 = IP1
    2            IF (IFP1 .LT. IP2) THEN
                   IFP2 = 2 * IFP1
                   THETA = ISIGN * 6.2831853071717959D0/ (IFP2/IP1)
                   WPR = -2.D0 * DSIN(0.5D0 * THETA) ** 2
                   WPI = DSIN (THETA)
                   WR = 1.D0
                   WI = 0.D0
                   DO 17 I3 =1, IFP1, IP1
                      DO 16 I1 = I3, I3 +IP1 - 2, 2
                           DO 15 I2 = I1, IP3, IFP2
                               K1 = I2
                               K2 = K1 + IFP1
                               TEMPR = SNGL(WR)*DATA(K2)-SNGL(WI)*DATA(K2+1)
                               TEMPI = SNGL(WR)*DATA(K2+1)+SNGL(WI)*DATA(K2)
                               DATA(K2) = DATA(K1) - TEMPR
                               DATA(K2 + 1) = DATA(K1 + 1) - TEMPI
                               DATA(K1) = DATA(K1) + TEMPR
                               DATA(K1 + 1) = DATA(K1 + 1) + TEMPI
   15                       CONTINUE
   16                  CONTINUE
                       WTEMP = WR
                       WR = WR * WPR - WI * WPI + WR
                       WI = WI * WPR + WTEMP * WPI + WI
   17               CONTINUE
                    IFP1 = IFP2
                    GO TO 2
                    ENDIF
                    NPREV = N * NPREV
   18 CONTINUE
      RETURN
      END
*
      SUBROUTINE FWT (NUMSAMPLS,DATARRAY,ENABLESCALE)
*
C     FAST WALSH TRANSFORM (TRANSLATED FROM D.S.D.'S VERSION IN "C")
C
C     NUMSAMPLS = NUMBER OF SEQUENCIES AND SAMPLING INTERVALS
C                 IN WALSH BASIS (INTEGER)
C
C     DATARRAY = INPUT/OUTPUT ARRAY (REAL)
```

```
C
C     ENABLESCALE = TRANSFORM SCALING ENABLE SWITCH
C
      INTEGER NUMSAMPLS
      REAL DATARRAY(1)
      LOGICAL ENABLESCALE
*
      CALL BITREVSORT (NUMSAMPLS,DATARRAY)
*
      CALL DECINSEQ (NUMSAMPLS,DATARRAY)
*
      IF (ENABLESCALE) CALL SCALE (NUMSAMPLS,DATARRAY)
      RETURN
      END
*
      SUBROUTINE BITREVSORT (NUMSAMPLS,DATARRAY)
*
C     BINARY BIT - REVERSAL SORT
*
      INTEGER NUMSAMPLS,NUMSD2,DIRECT,REVERSED,DATAD,DATAR,OFFSET
      REAL DATARRAY(1),SCRATCH
*
      NUMSD2 = NUMSAMPLS / 2
      REVERSED = 0
      DO 30 DIRECT = 0, NUMSAMPLS - 1
         IF (DIRECT .GE. REVERSED) GO TO 10
            DATAD = 1 + DIRECT
            DATAR = 1 + REVERSED
            SCRATCH = DATARRAY(DATAD)
            DATARRAY(DATAD) = DATARRAY(DATAR)
            DATARRAY(DATAR) = SCRATCH
   10    OFFSET = NUMSD2
   20    IF((OFFSET .GT. REVERSED) .OR. (OFFSET .LT. 2)) GO TO 30
            REVERSED = REVERSED - OFFSET
            OFFSET = OFFSET / 2
            GO TO 20
   30 REVERSED = REVERSED + OFFSET
      RETURN
      END
*
      SUBROUTINE DECINSEQ (NUMSAMPLS,DATARRAY)
*
C     DECIMATION IN SEQUENCE
*
      INTEGER NUSAMPLS,LOG2NUMSAMPLS,MSB,BLOCKSIZE,STAGE
      INTEGER NODE1START,NODE2START,NODE1,NODE2
      LOGICAL TOGGLE
      REAL DATARRAY(1),SCRATCH
*
      LOG2NUMSAMPLS = 0
      MSB = NUMSAMPLS
   10 MSB = MSB / 2
      IF (MSB .EQ. 0) GO TO 20
         LOG2NUMSAMPLS = LOG2NUMSAMPLS + 1
         GO TO 10
   20 BLOCKSIZE = NUMSAMPLS / 2
      DO 80 STAGE = 1, LOG2NUMSAMPLS
         TOGGLE = .FALSE.
         NODE1START = 0
   30    IF (NODE1START .GE. NUMSAMPLS) GO TO 80
```

```
            NODE2START = NODE1START + BLOCKSIZE
            NODE2 = NODE2START
            IF (.NOT. TOGGLE) GO TO 50
               DO 40 NODE1 = NODE1START,NODE2START - 1
                  SCRATCH = DATARRAY(NODE1 + 1)
                  DATARRAY(NODE1+1)=DATARRAY(NODE1+1)-DATARRAY(NODE2+1)
                  DATARRAY(NODE2+1) = DATARRAY(NODE2+1) + SCRATCH
   40          NODE2 = NODE2 + 1
               GO TO 70
   50          DO 60 NODE1 = NODE1START,NODE2START - 1
                  SCRATCH = DATARRAY(NODE1+1)
                  DATARRAY(NODE1+1)=DATARRAY(NODE1+1)+DATARRAY(NODE2+1)
                  DATARRAY(NODE2+1) = SCRATCH - DATARRAY(NODE2+1)
   60          NODE2 = NODE2 + 1
   70       TOGGLE = .NOT. TOGGLE
            NODE1START = NODE2
            GO TO 30
   80    BLOCKSIZE = BLOCKSIZE / 2
         RETURN
         END
*
         SUBROUTINE SCALE (NUMSAMPLS,DATARRAY)
*
C        WALSH TRANSFORM SCALING
*
         INTEGER NUMSAMPLS,I
         REAL DATARRAY(1),SCALEFACTOR
*
         SCALEFACTOR = 1.0 /NUMSAMPLS
         DO 10 I = 1, NUMSAMPLS
           DATARRAY(I) = DATARRAY(I) * SCALEFACTOR
   10    CONTINUE
         RETURN
         CLOSE (11)
         END
```

APPENDIX B

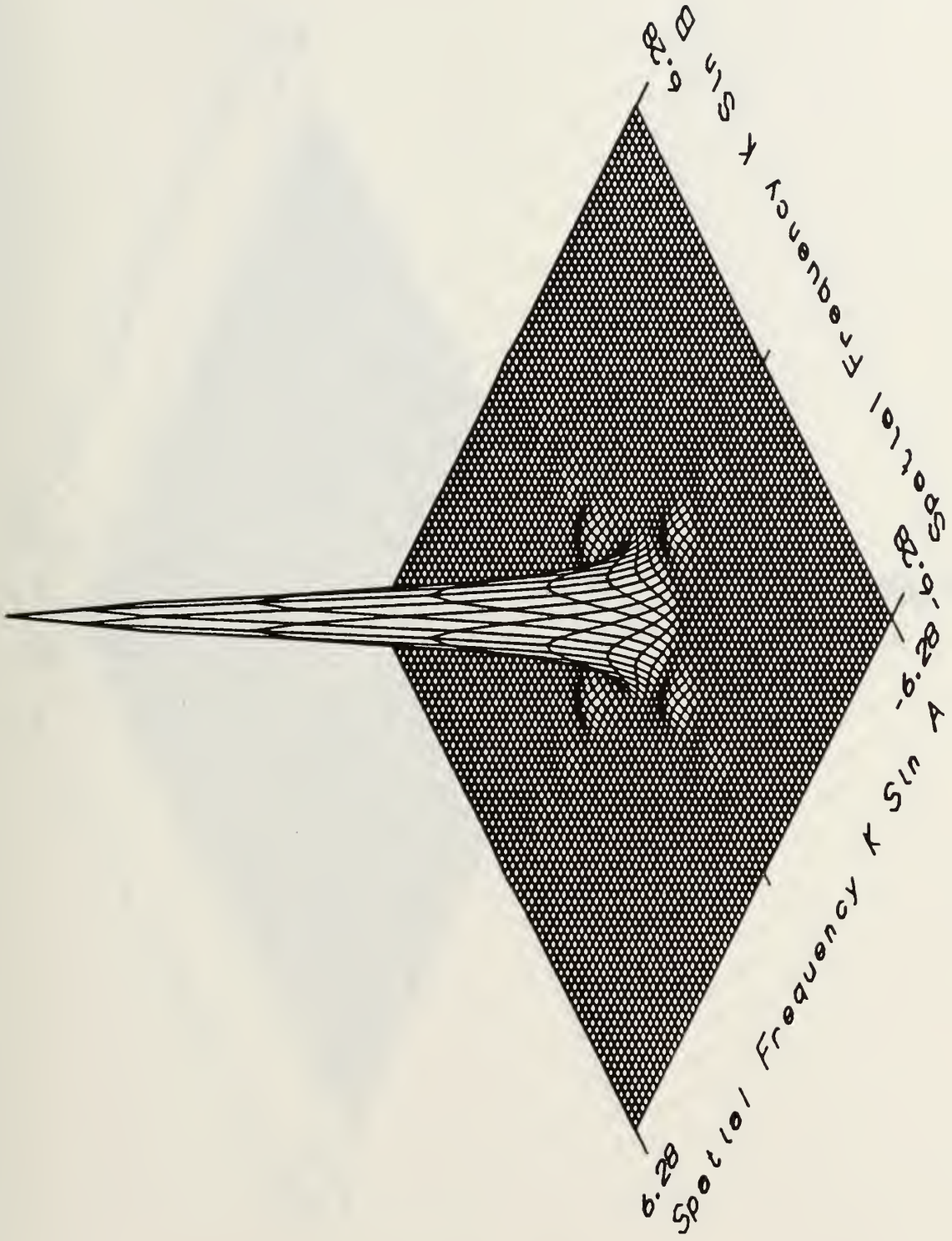

Figure B-1 Diffraction Pattern for Sequency Zero

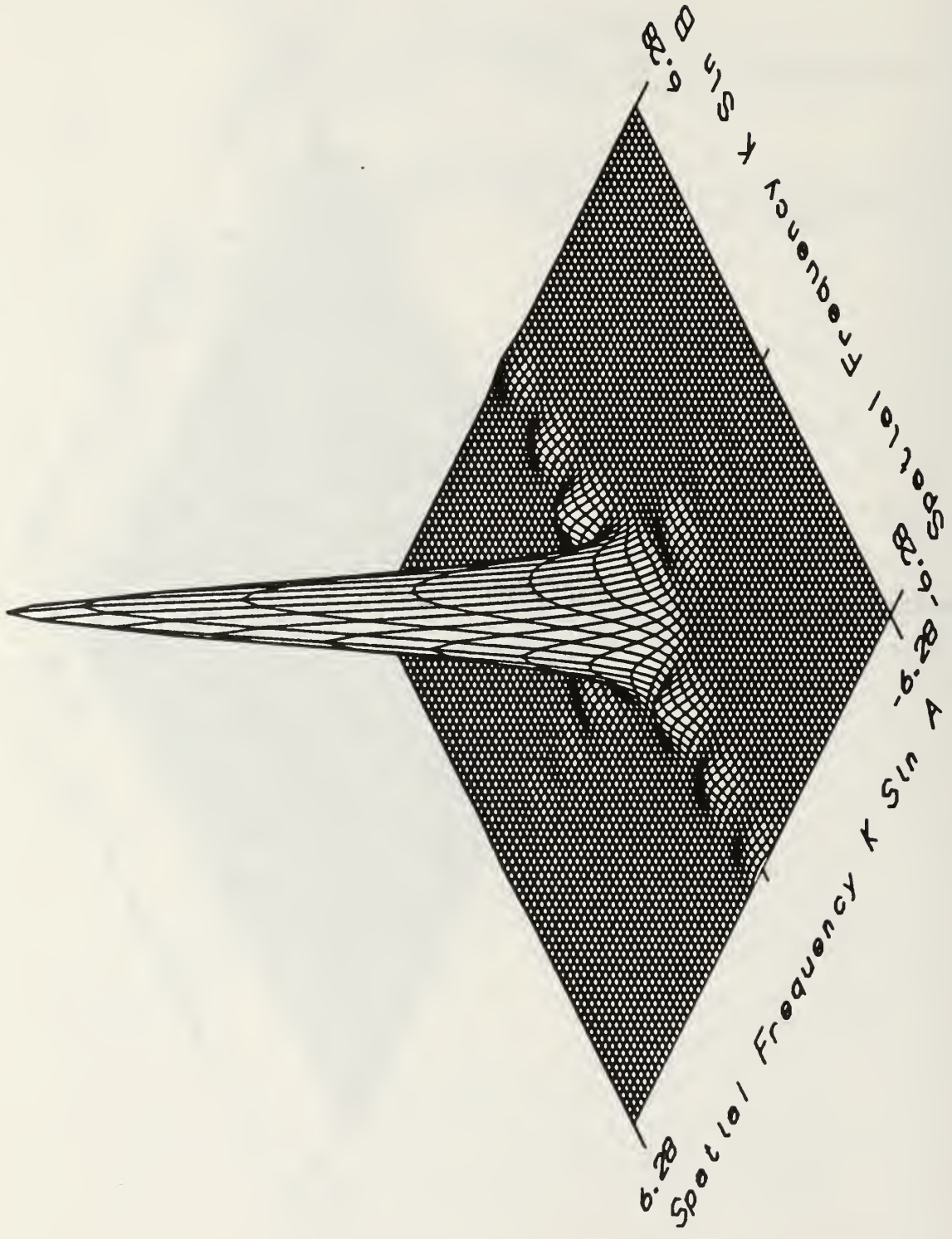

Figure B-2 Diffraction Pattern for Sequency One

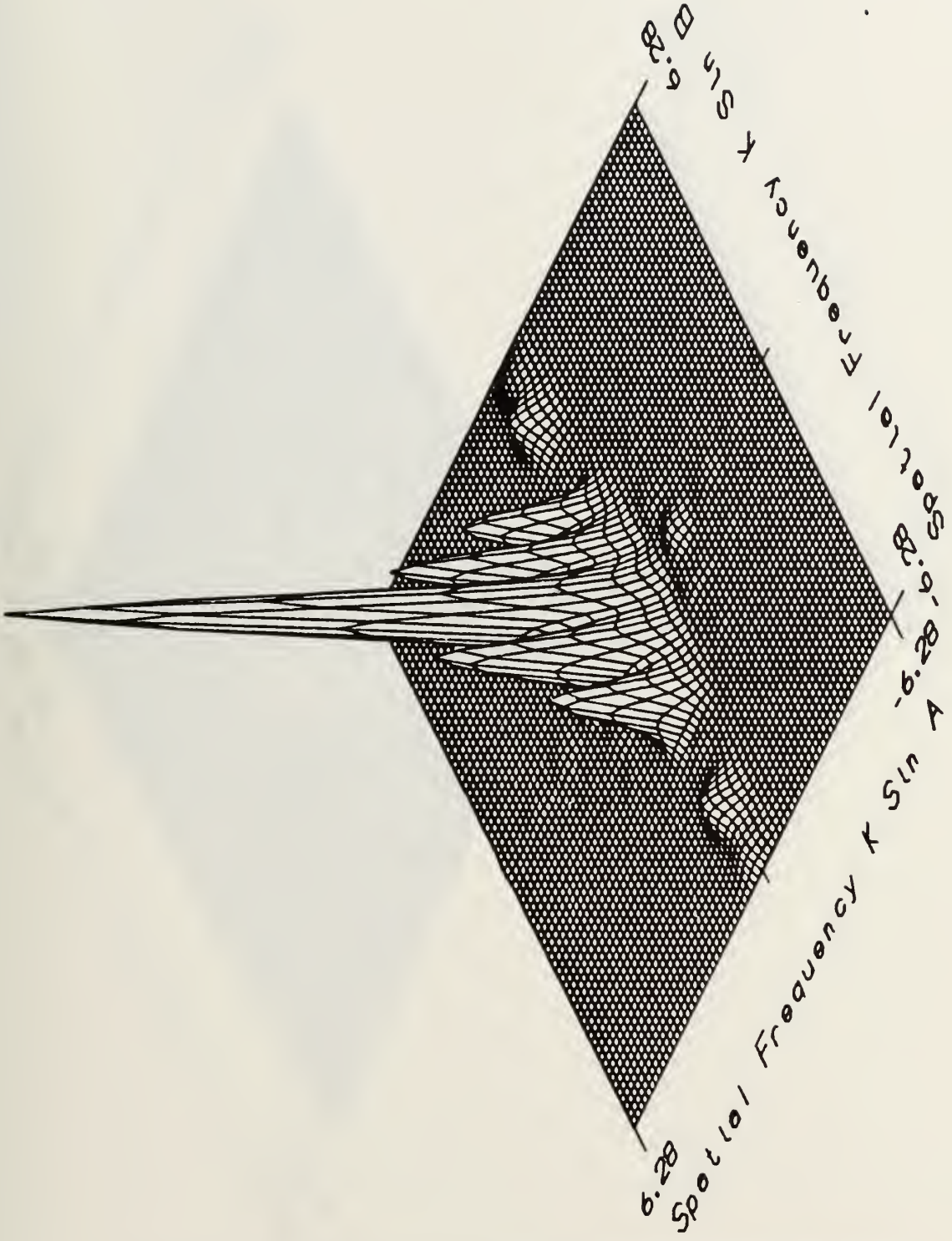

Figure B-3 Diffraction Pattern for Sequency Two

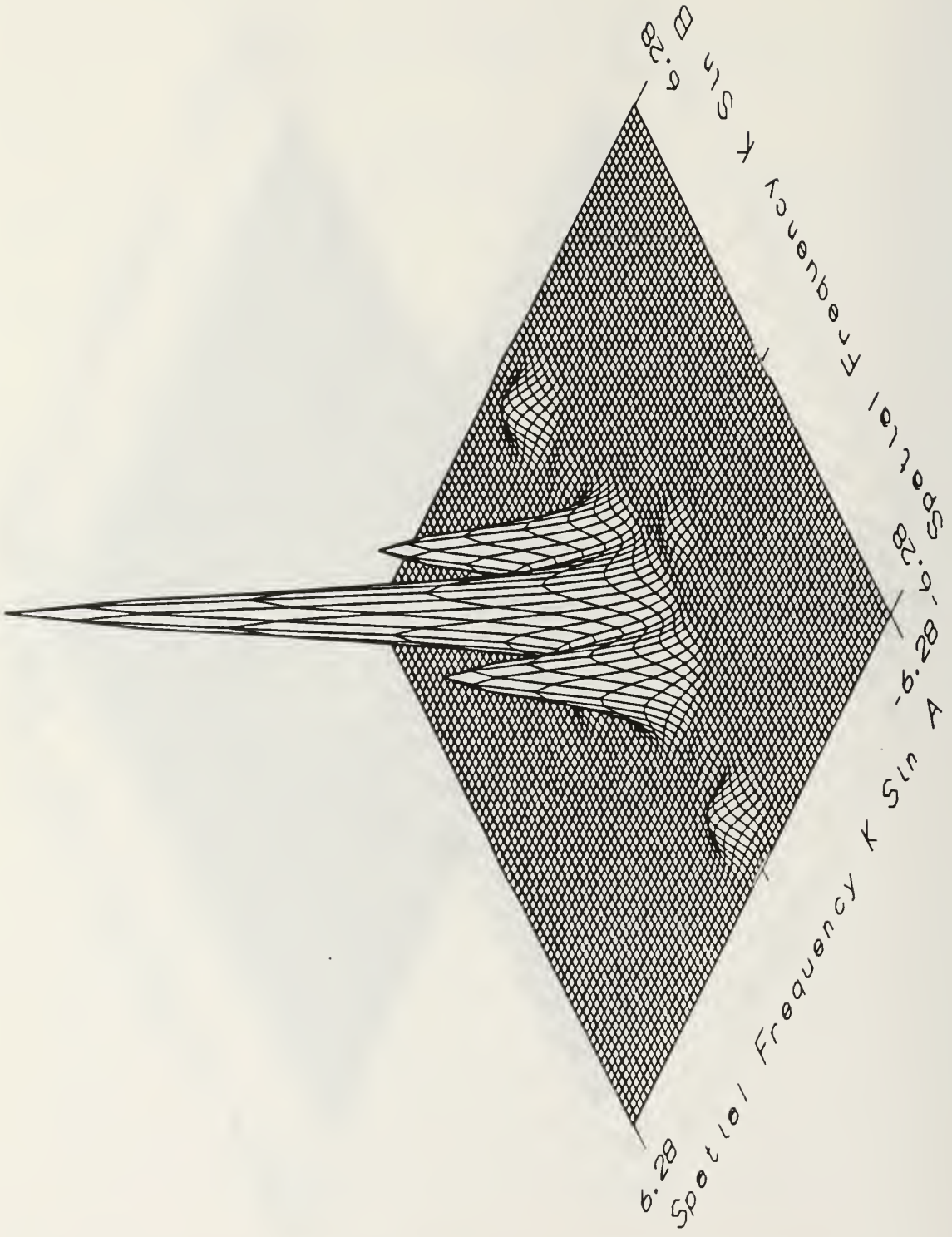

Figure B-4 Diffraction Pattern for Sequency Three

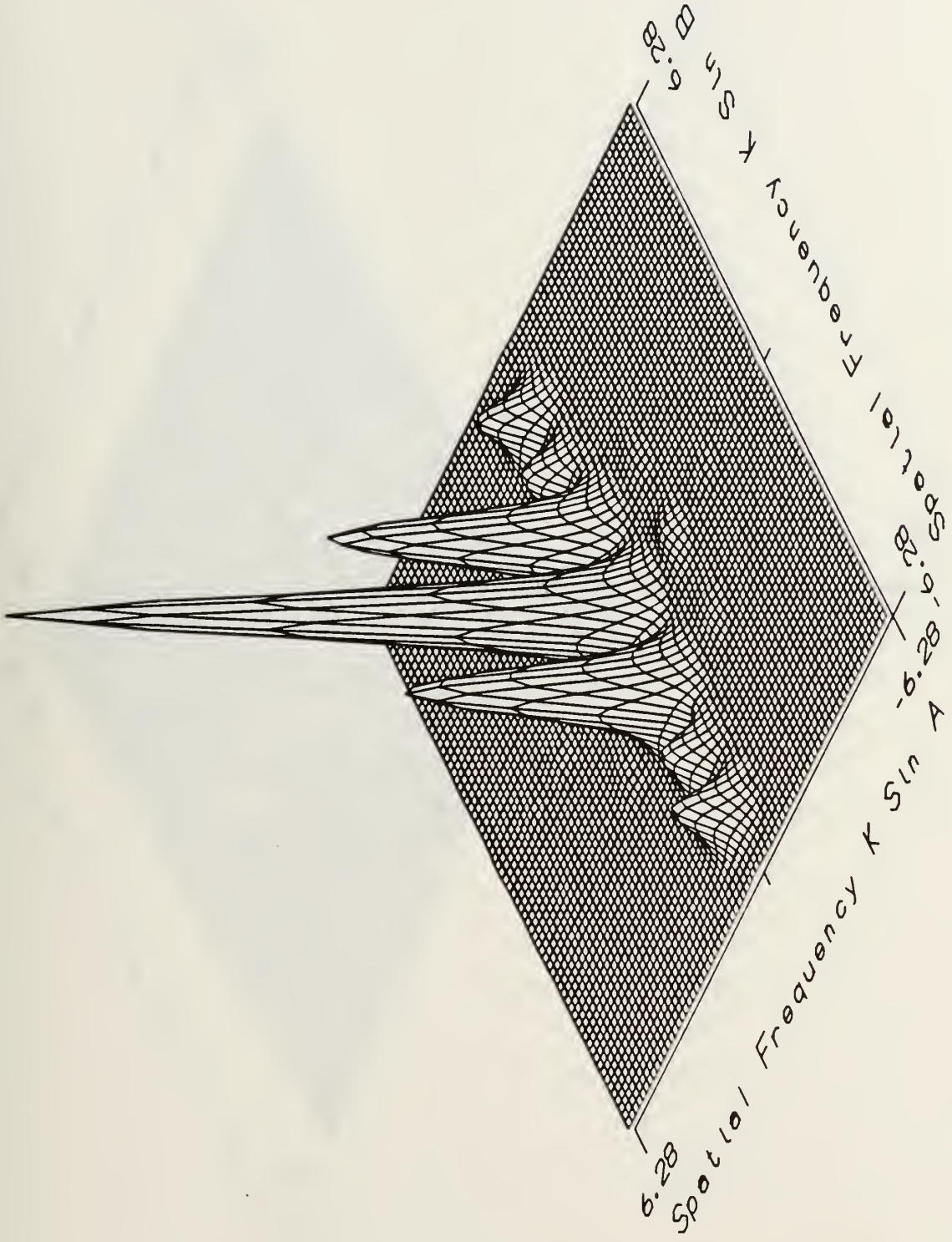

Figure B-5 Diffraction Pattern for Sequency Four

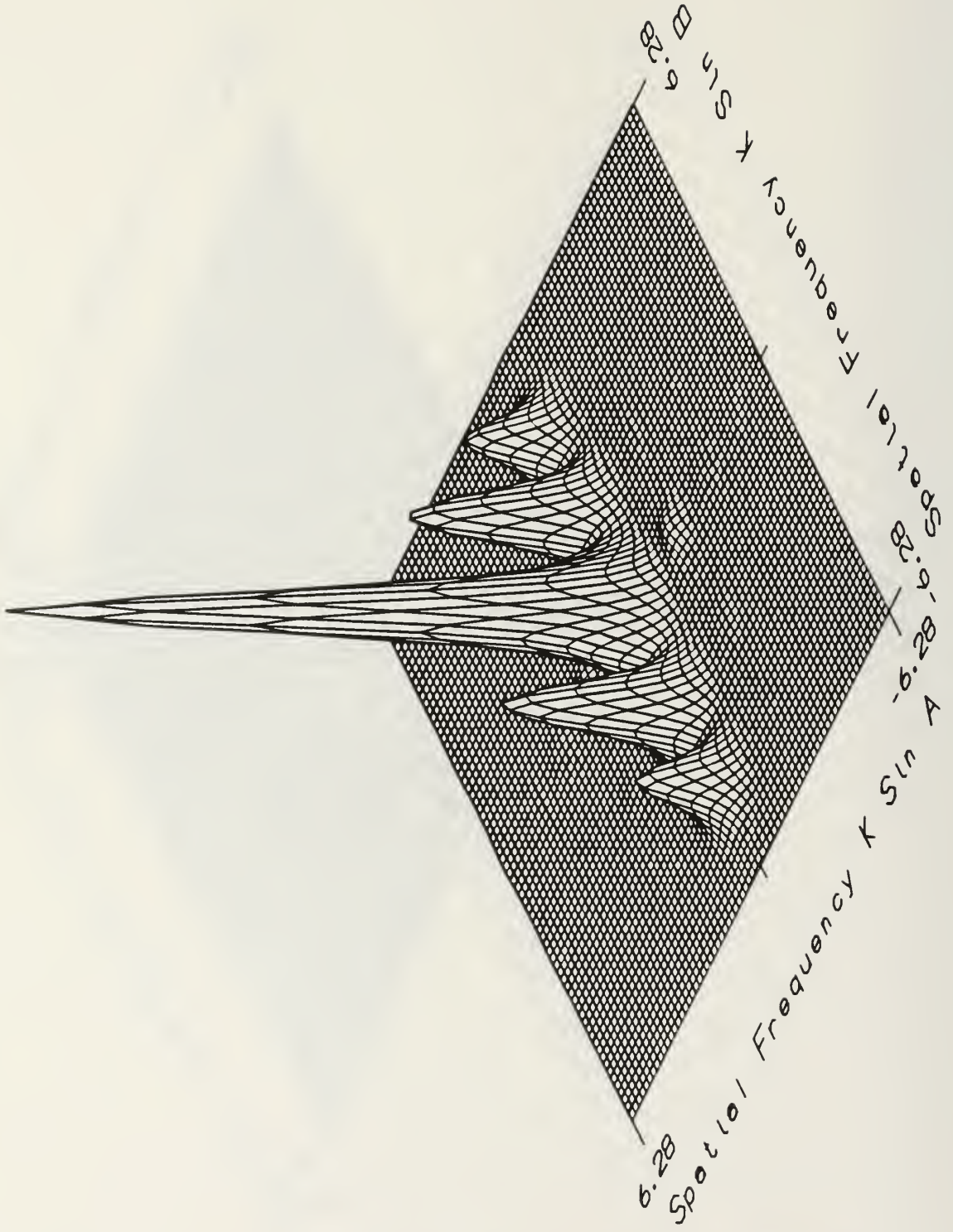

Figure B-6 Diffraction Pattern for Sequency Five

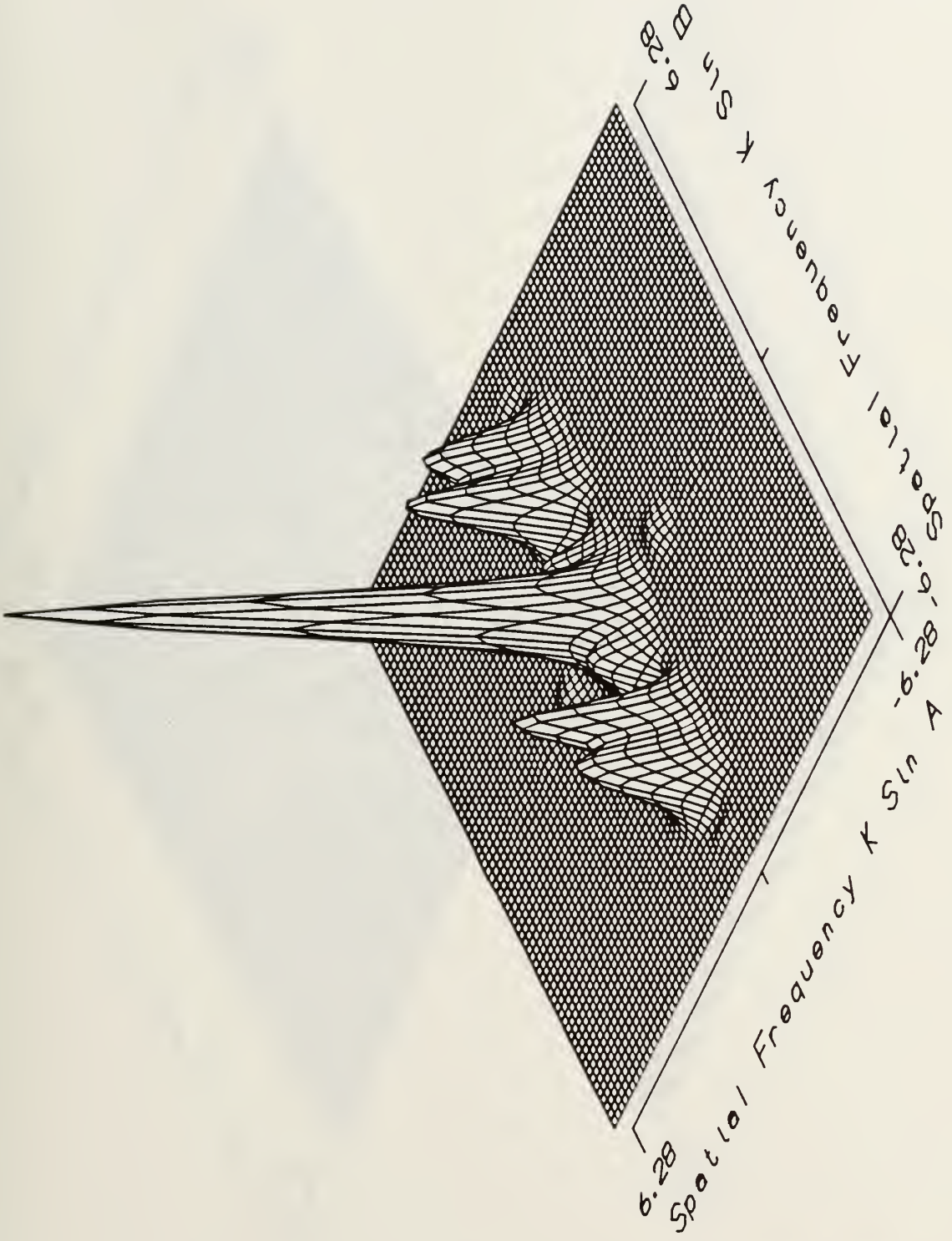

Figure B-7 Diffraction Pattern for Sequency Six

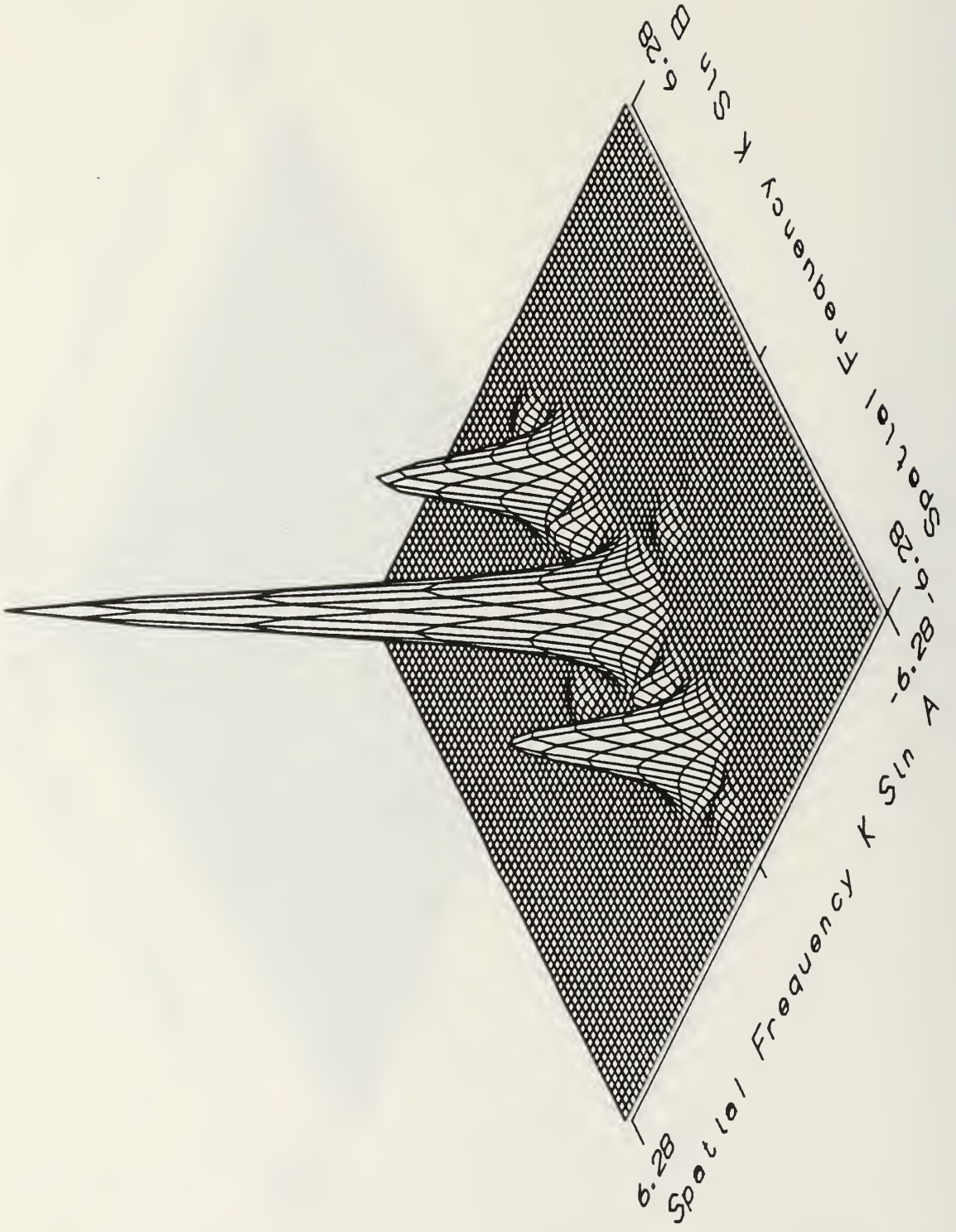

Figure B-8 Diffraction Pattern for Sequency Seven

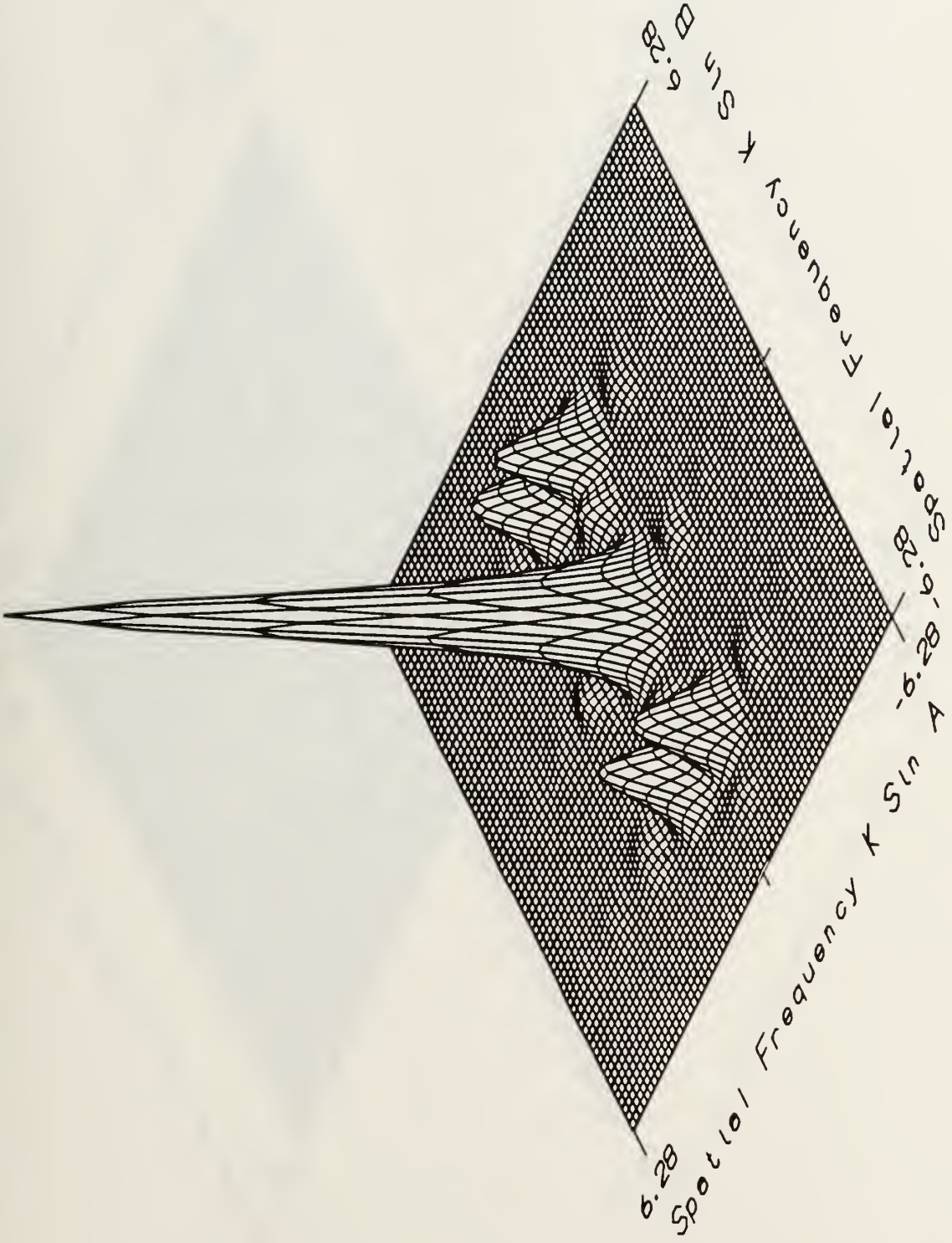

Figure B-9 Diffraction Pattern for Sequency Eight

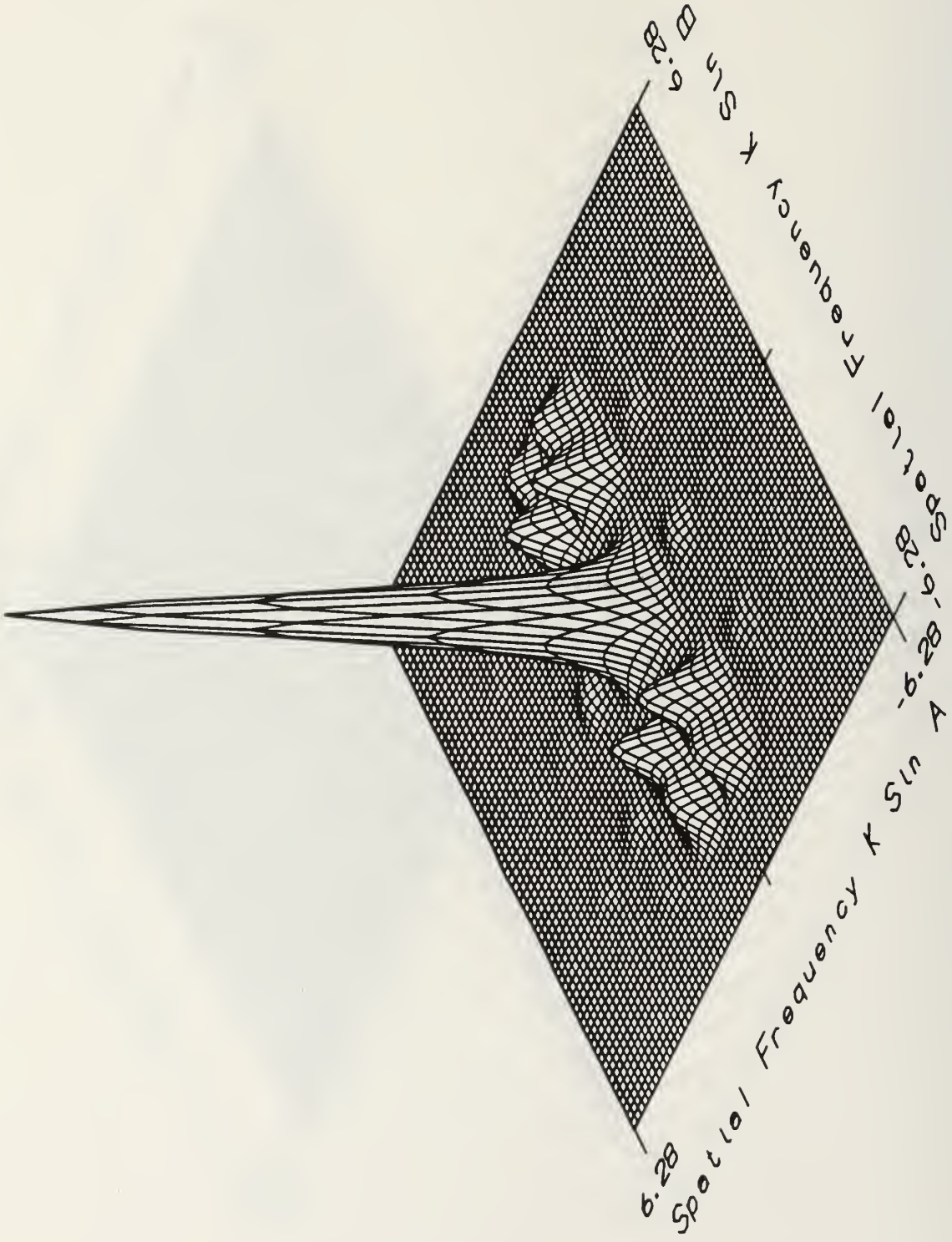

Figure B-10 Diffraction Pattern for Sequency Nine

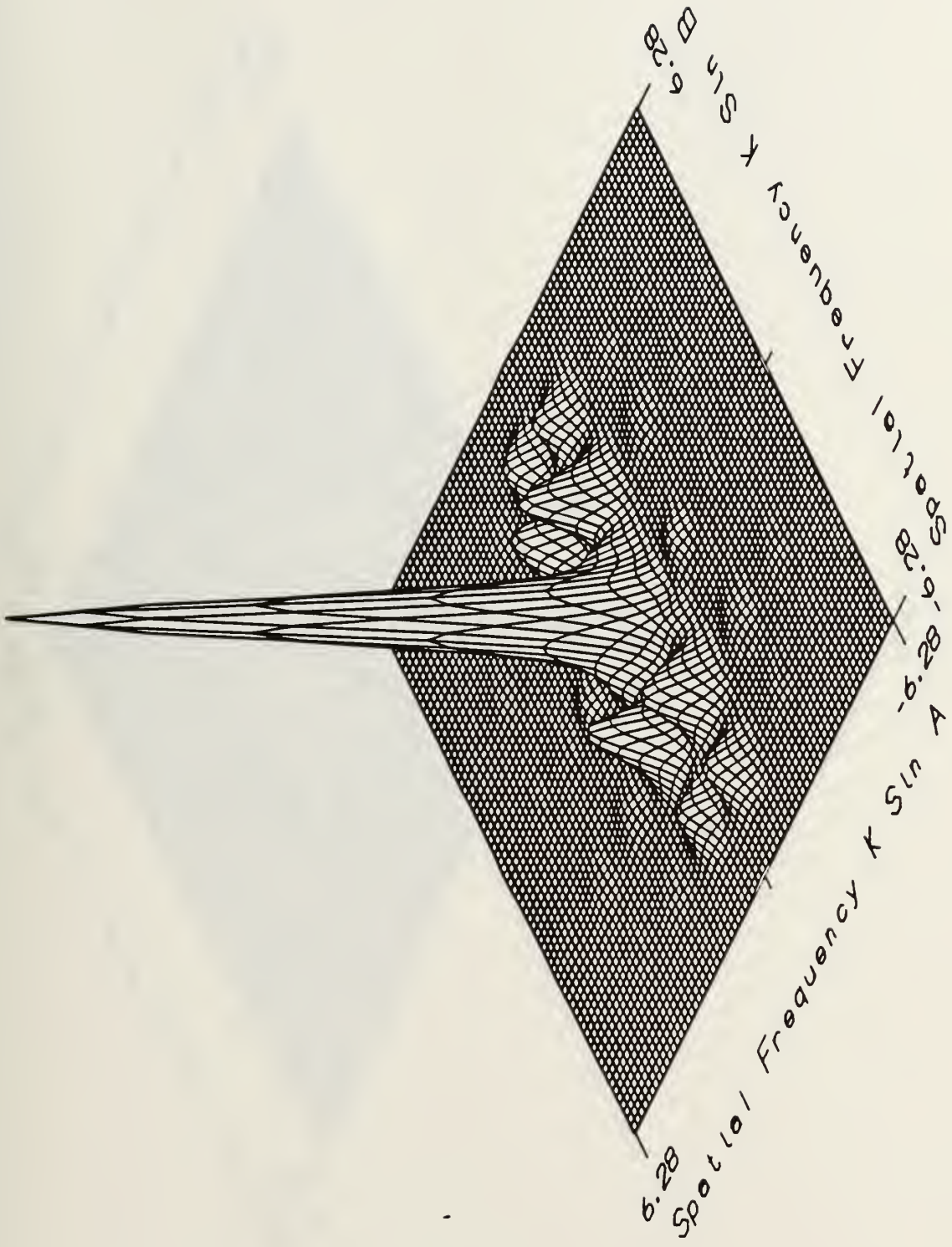

Figure B-11 Diffraction Pattern for Sequency Ten

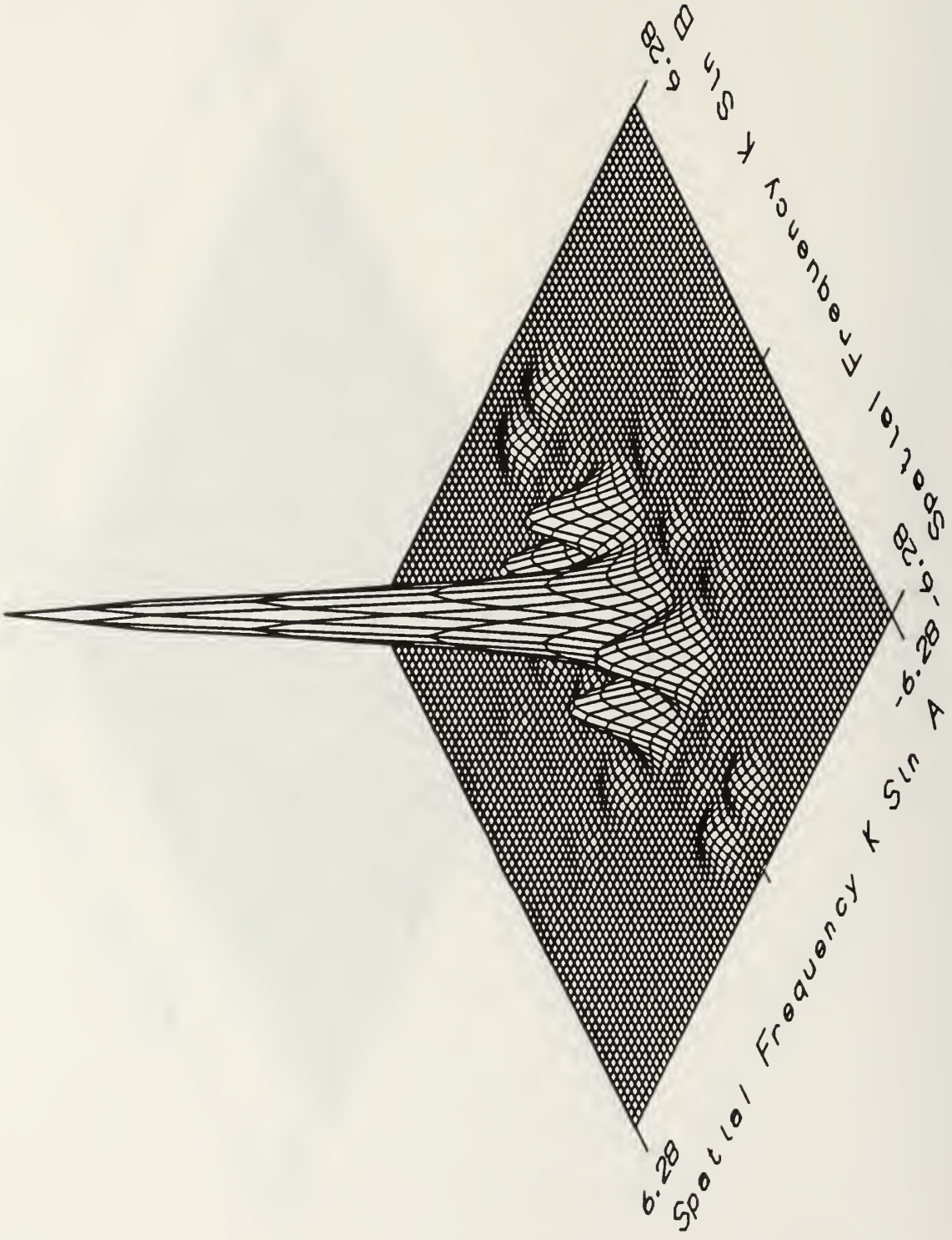

Figure B-12 Diffraction Pattern for Sequency 11

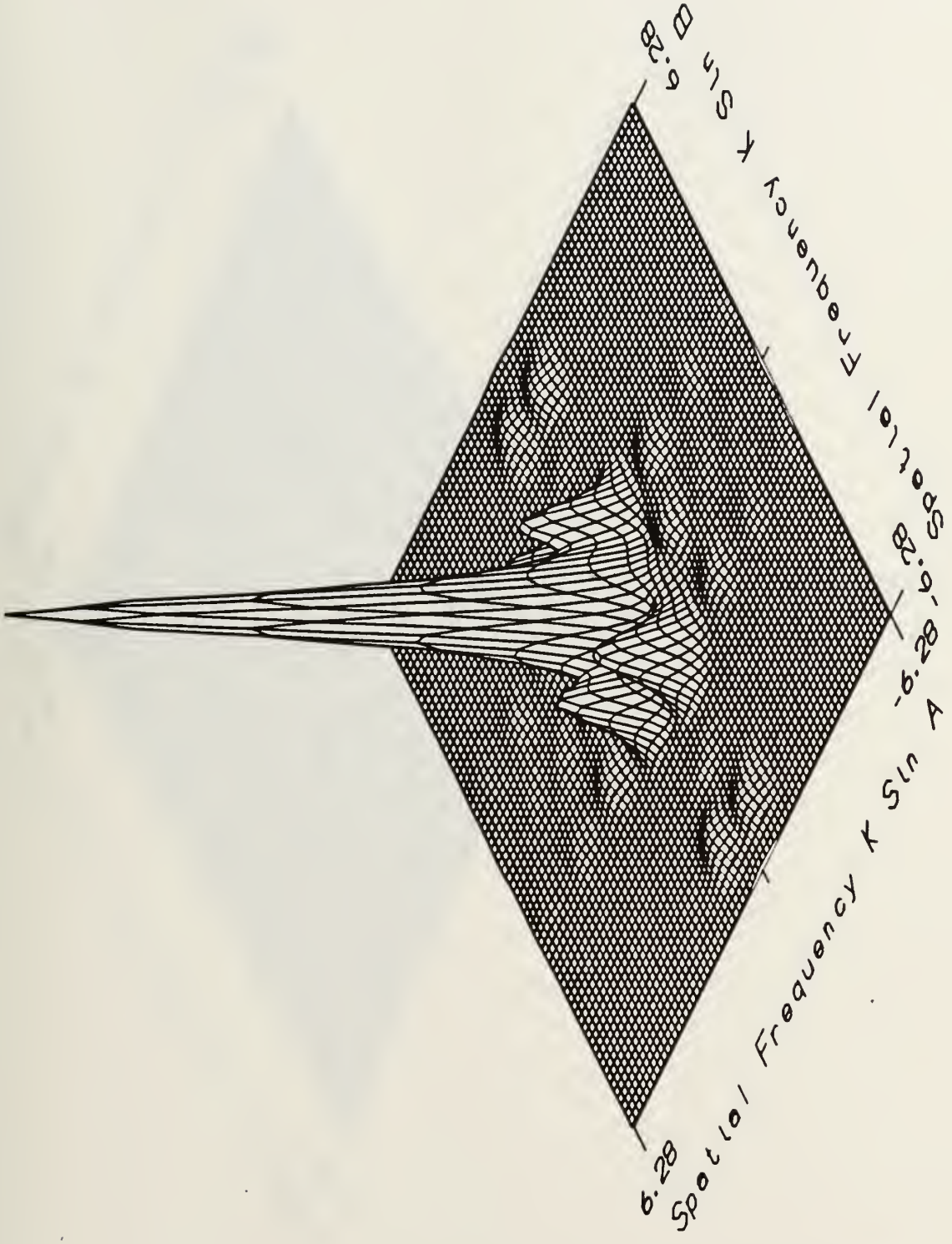

Figure B-13 Diffraction Pattern for Sequency 12

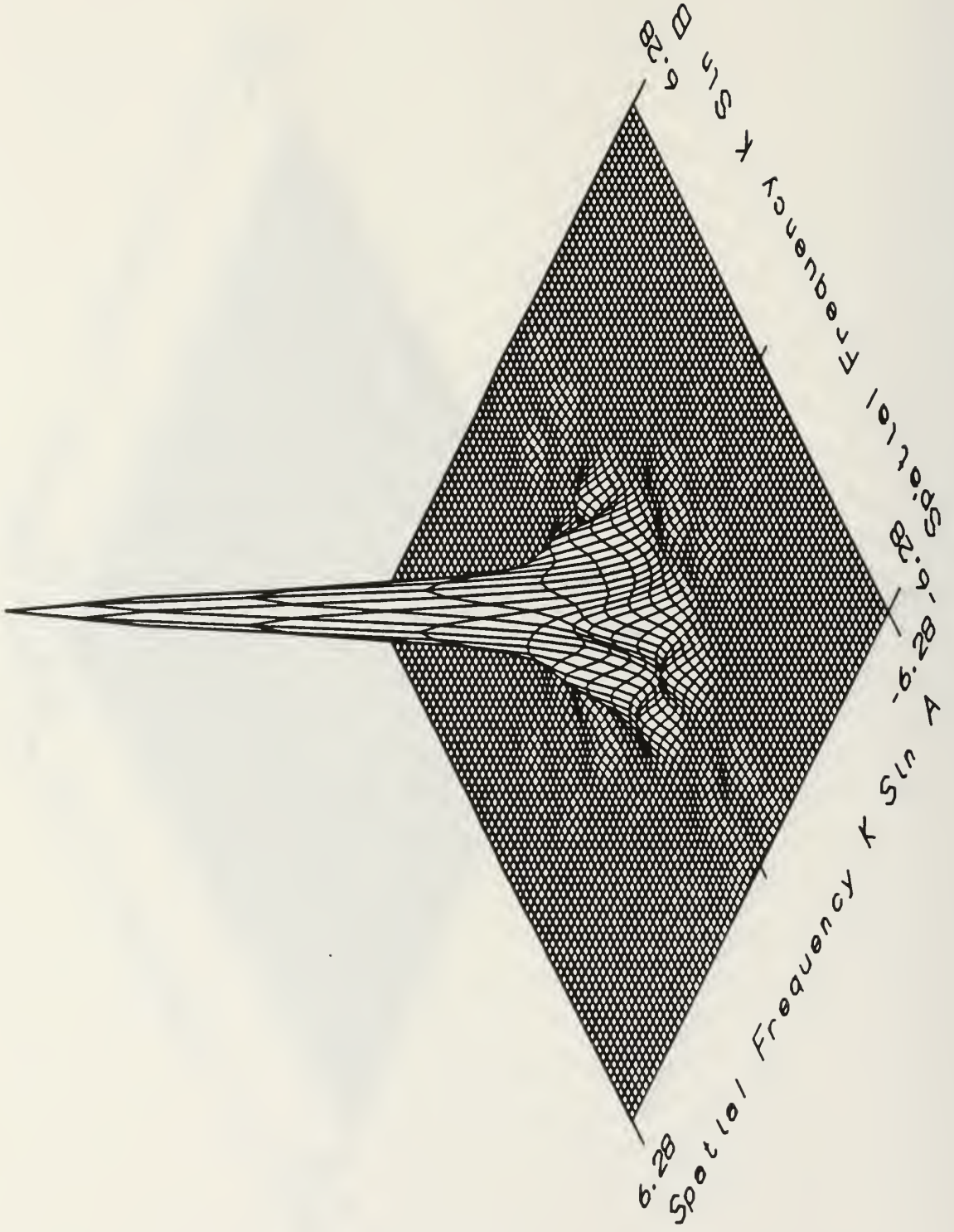

Figure B-14 Diffraction Pattern for Sequency 13

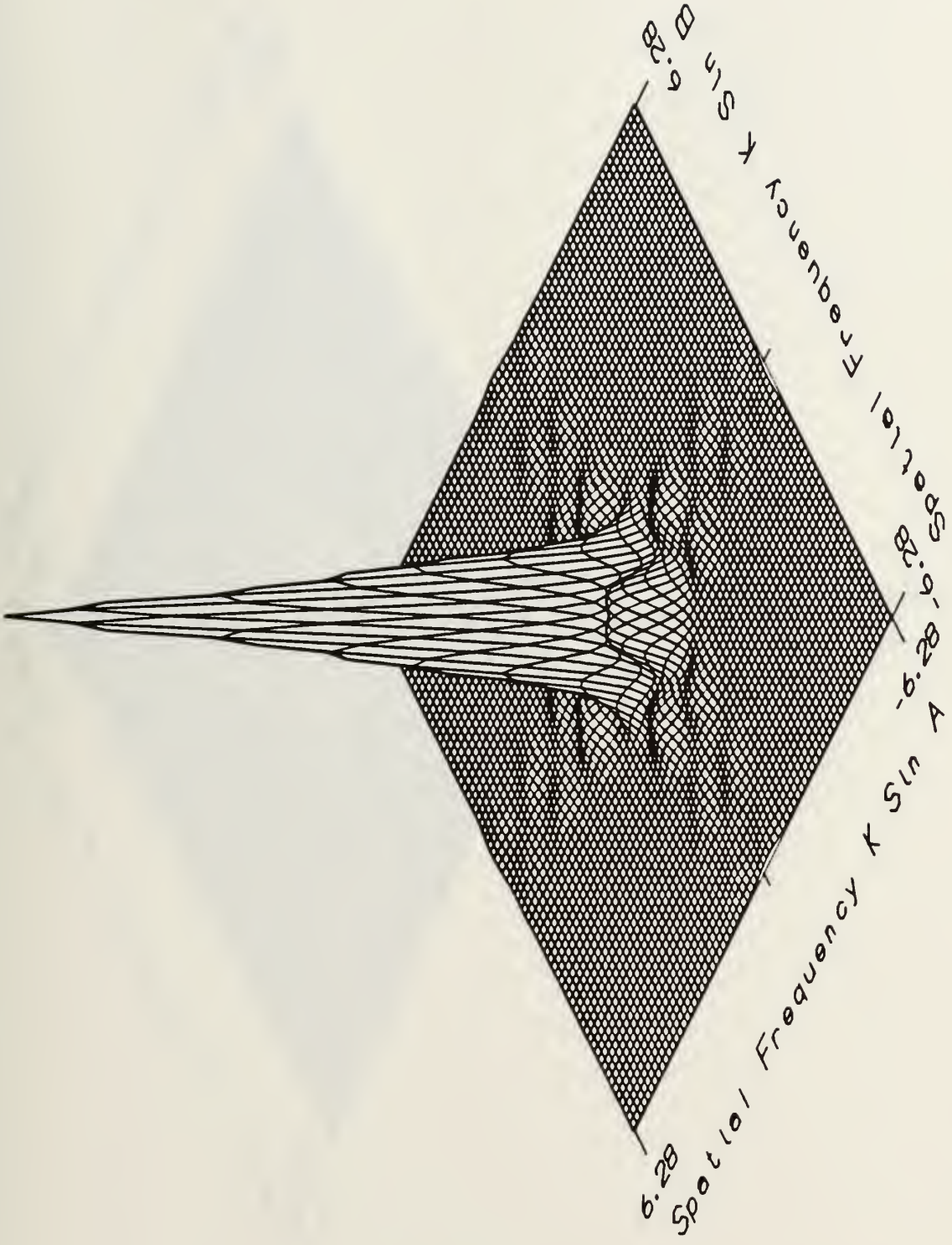

Figure B-15 Diffraction Pattern for Sequency 14

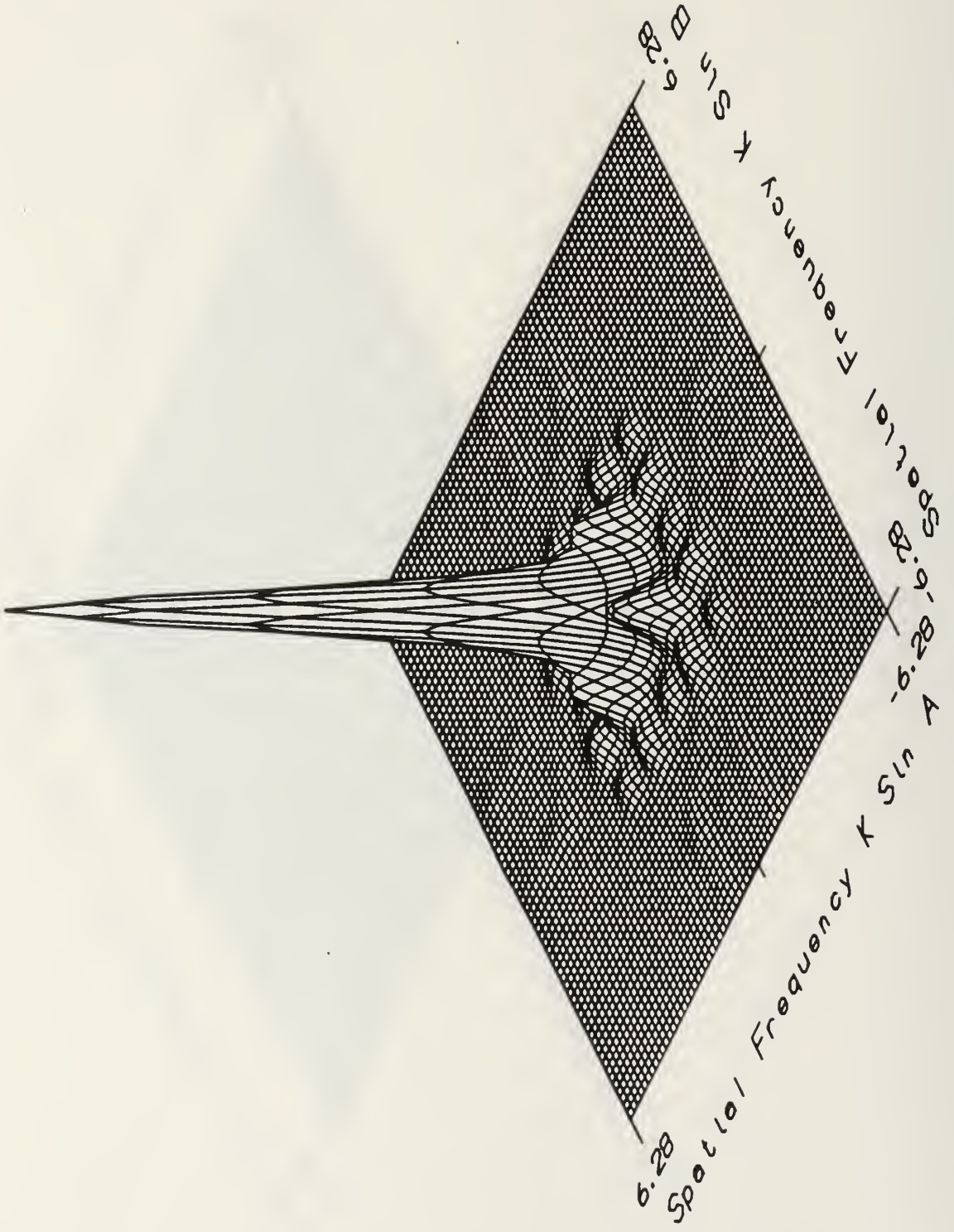

Figure B-16 Diffraction Pattern for Sequency 18

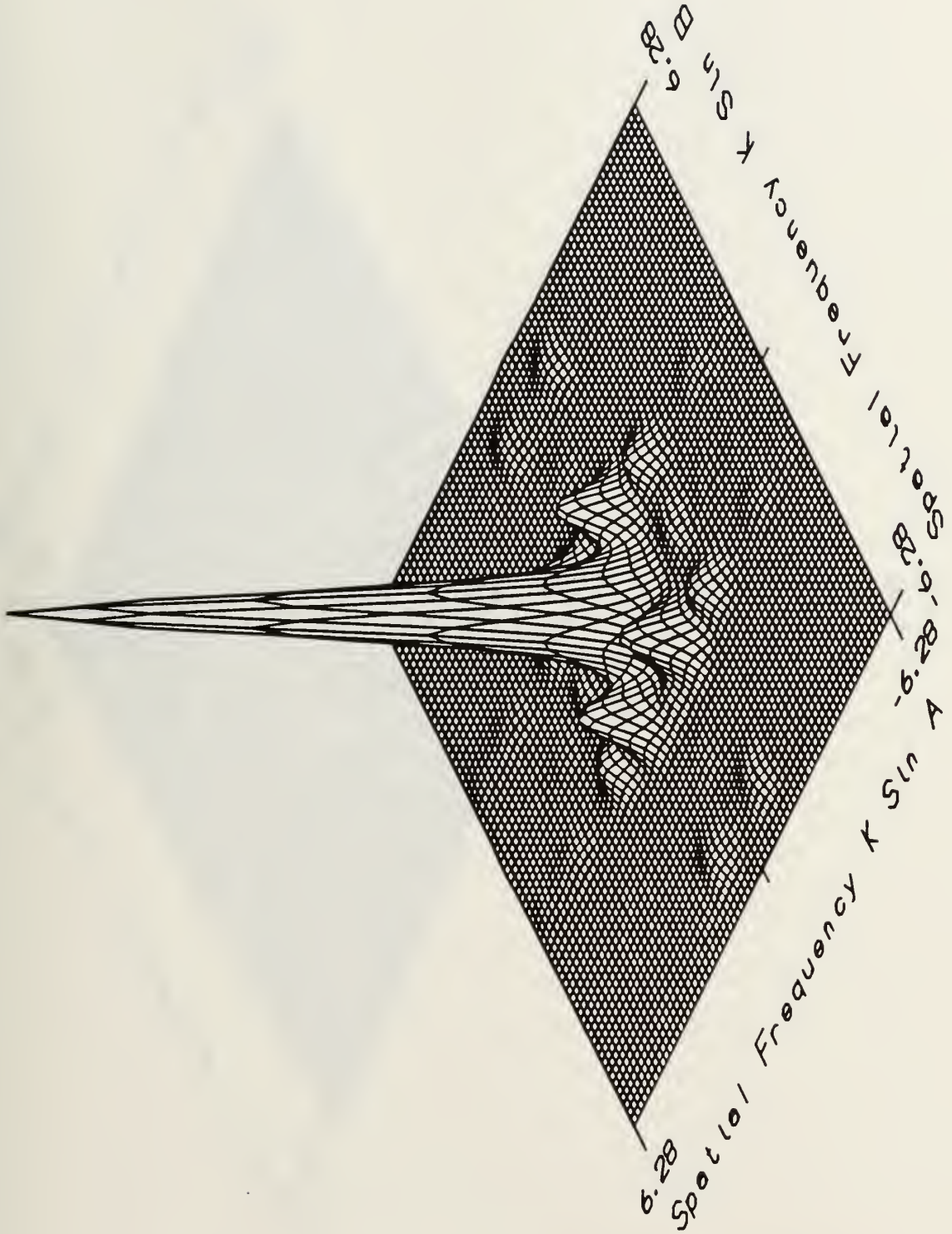

Figure B-17 Diffraction Pattern for Sequency 19

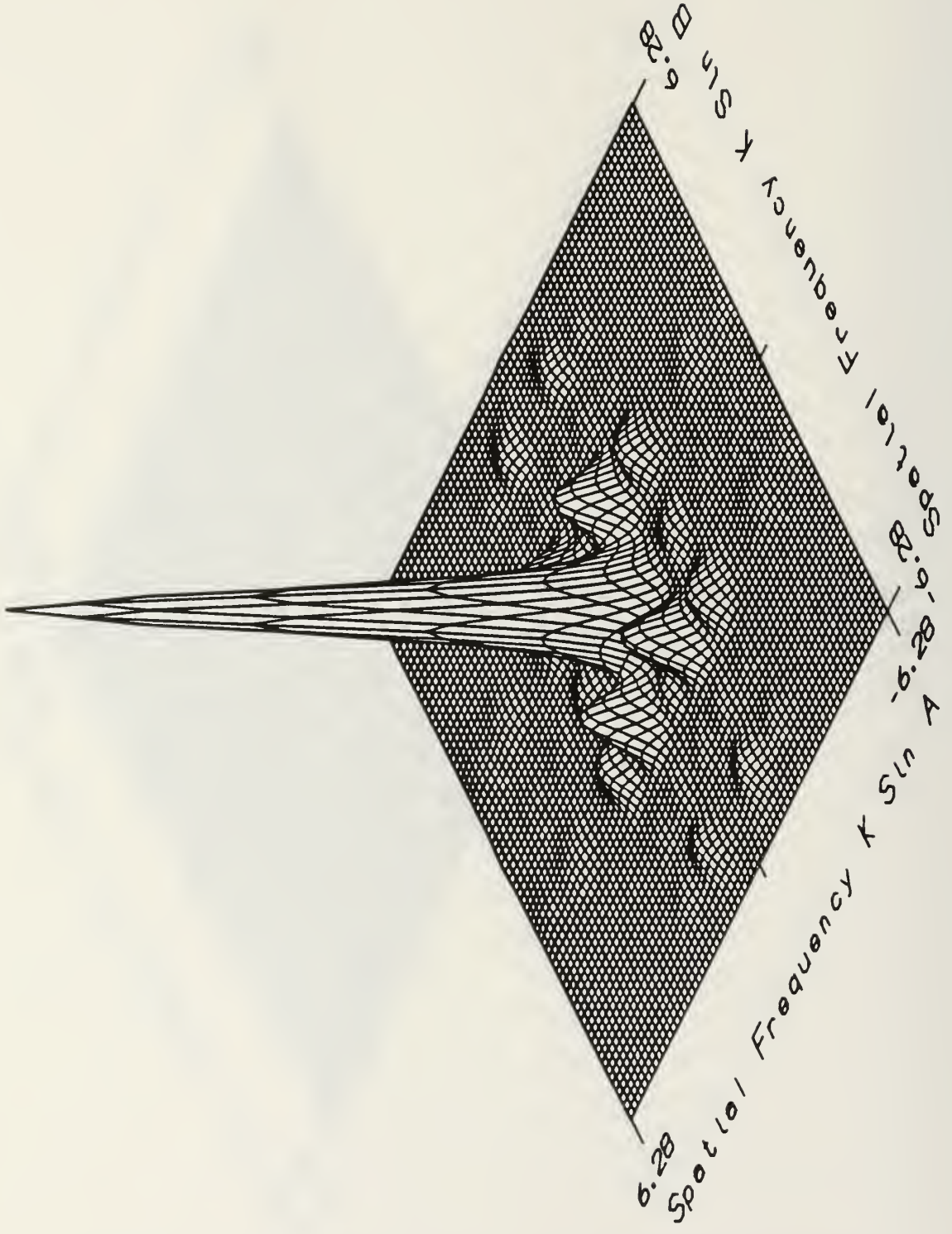

Figure B-18 Diffraction Pattern for Sequency 20

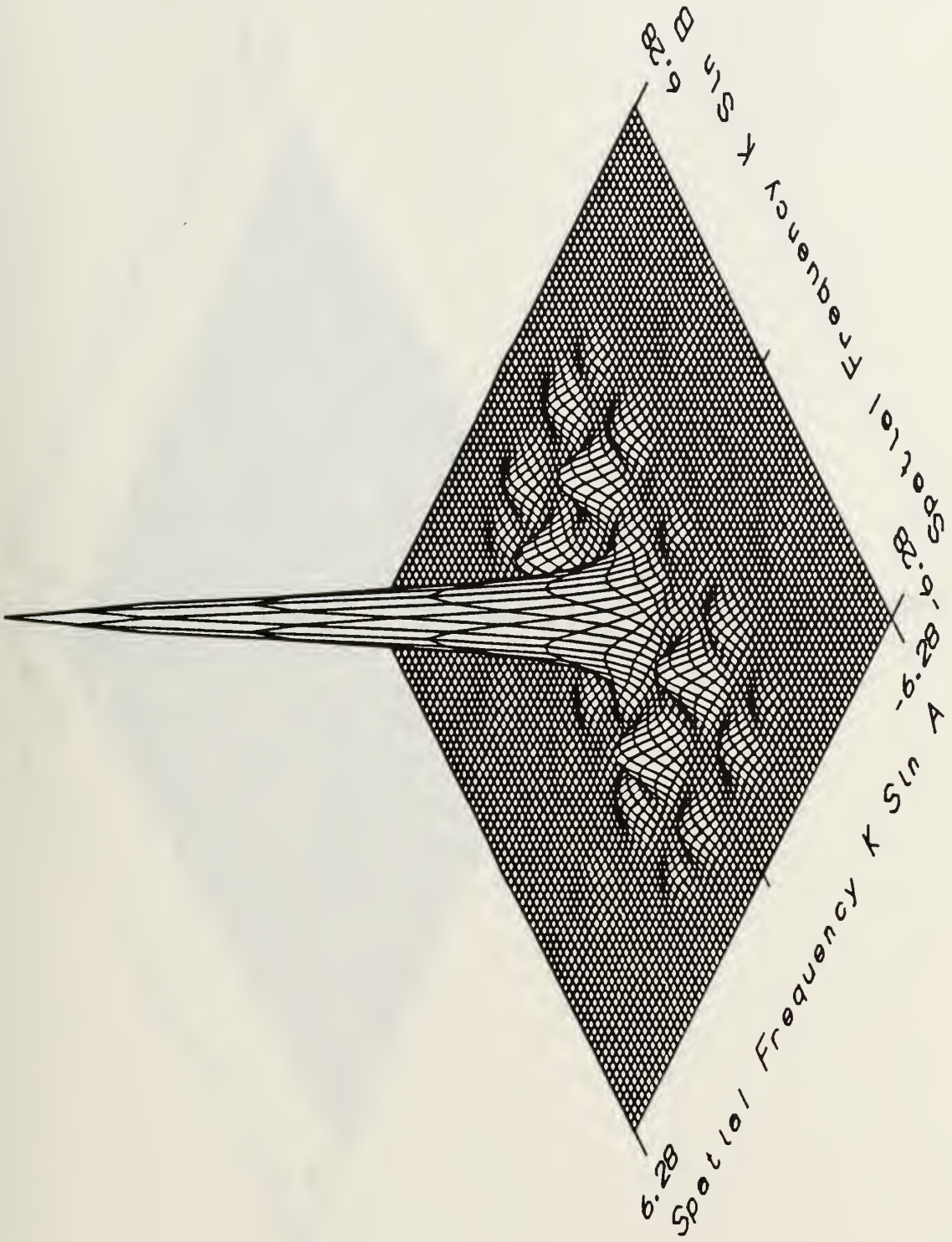

Figure B-19 Diffraction Pattern for Sequency 21

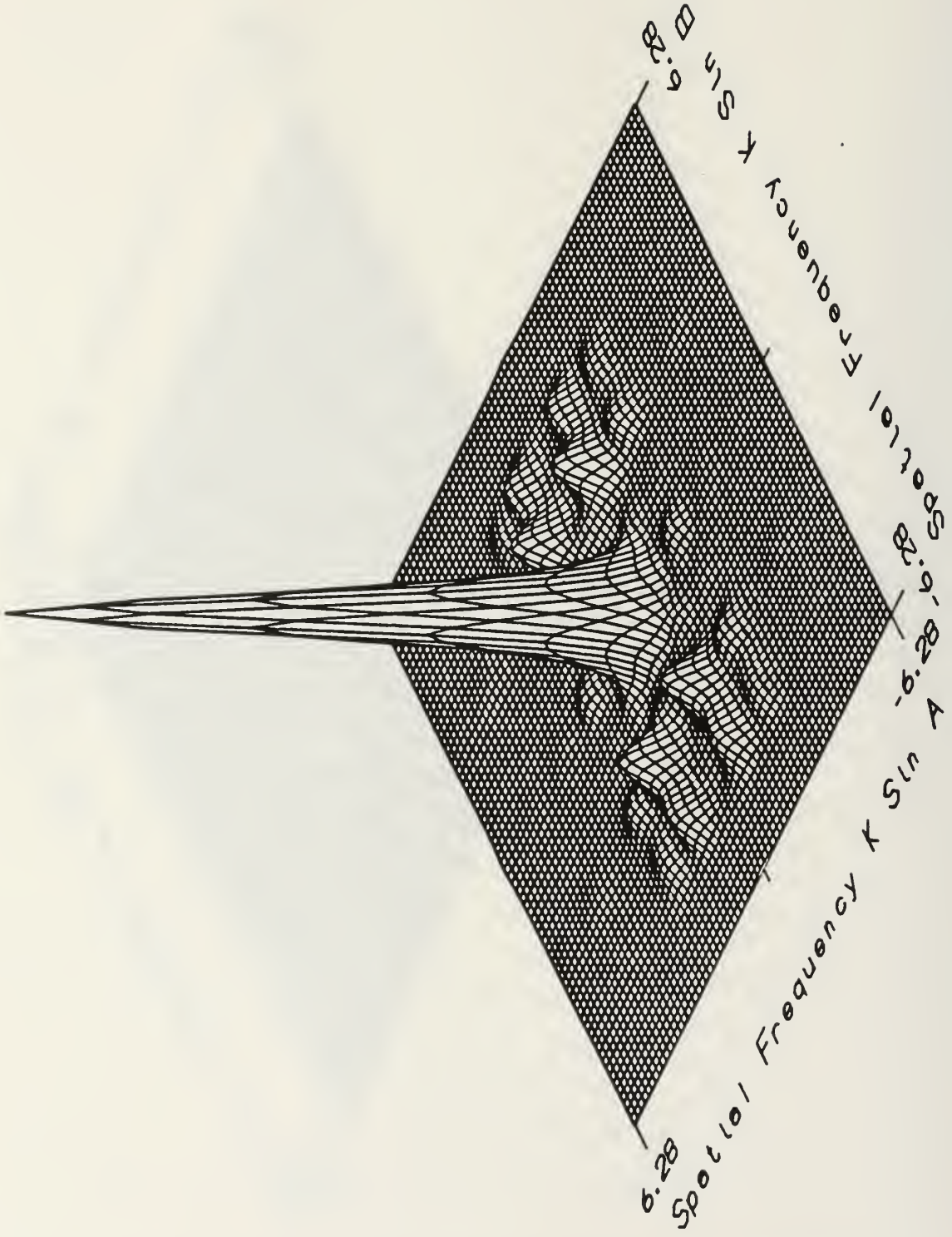

Figure B-20 Diffraction Pattern for Sequency 22

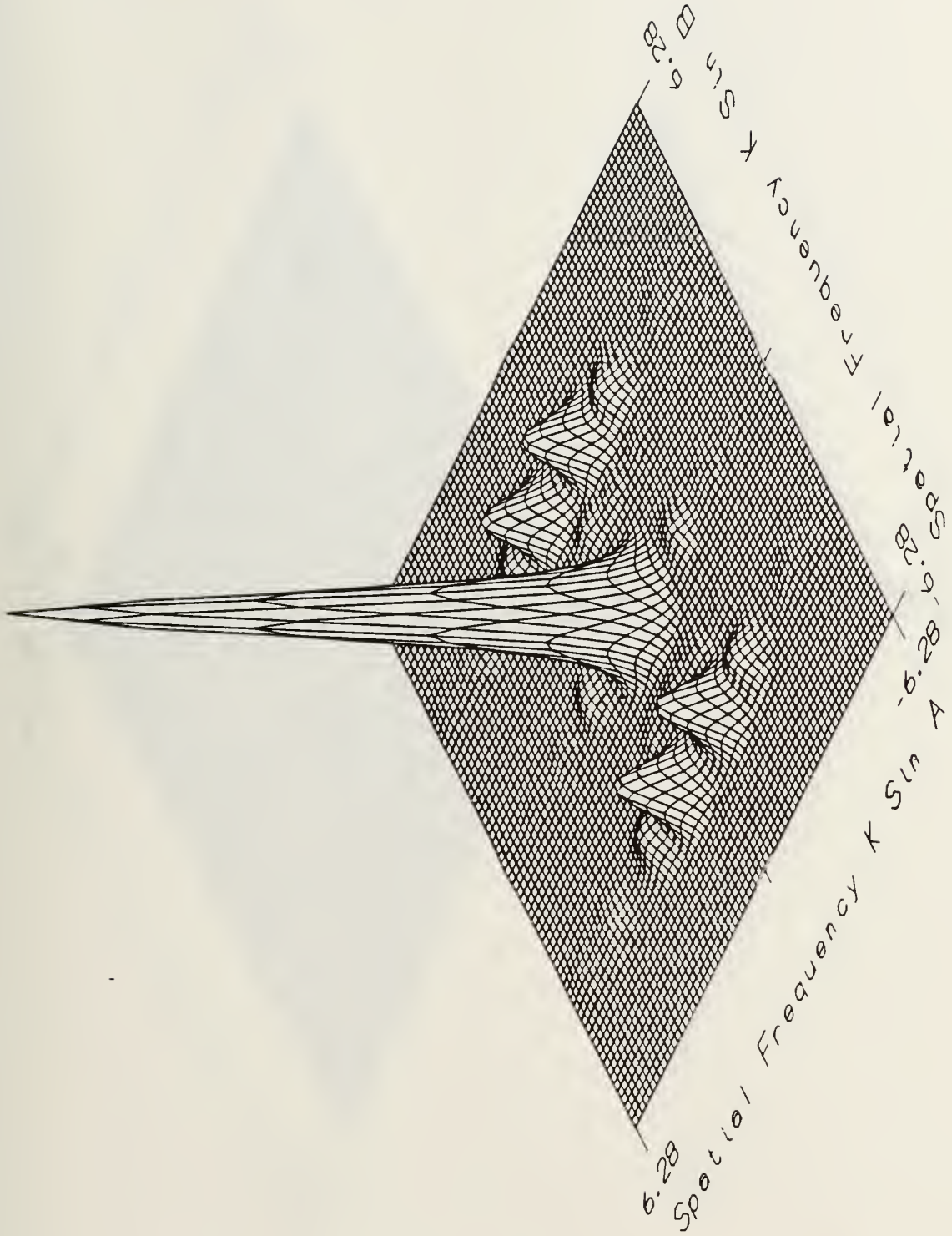

Figure B-21 Diffraction Pattern for Sequency 23

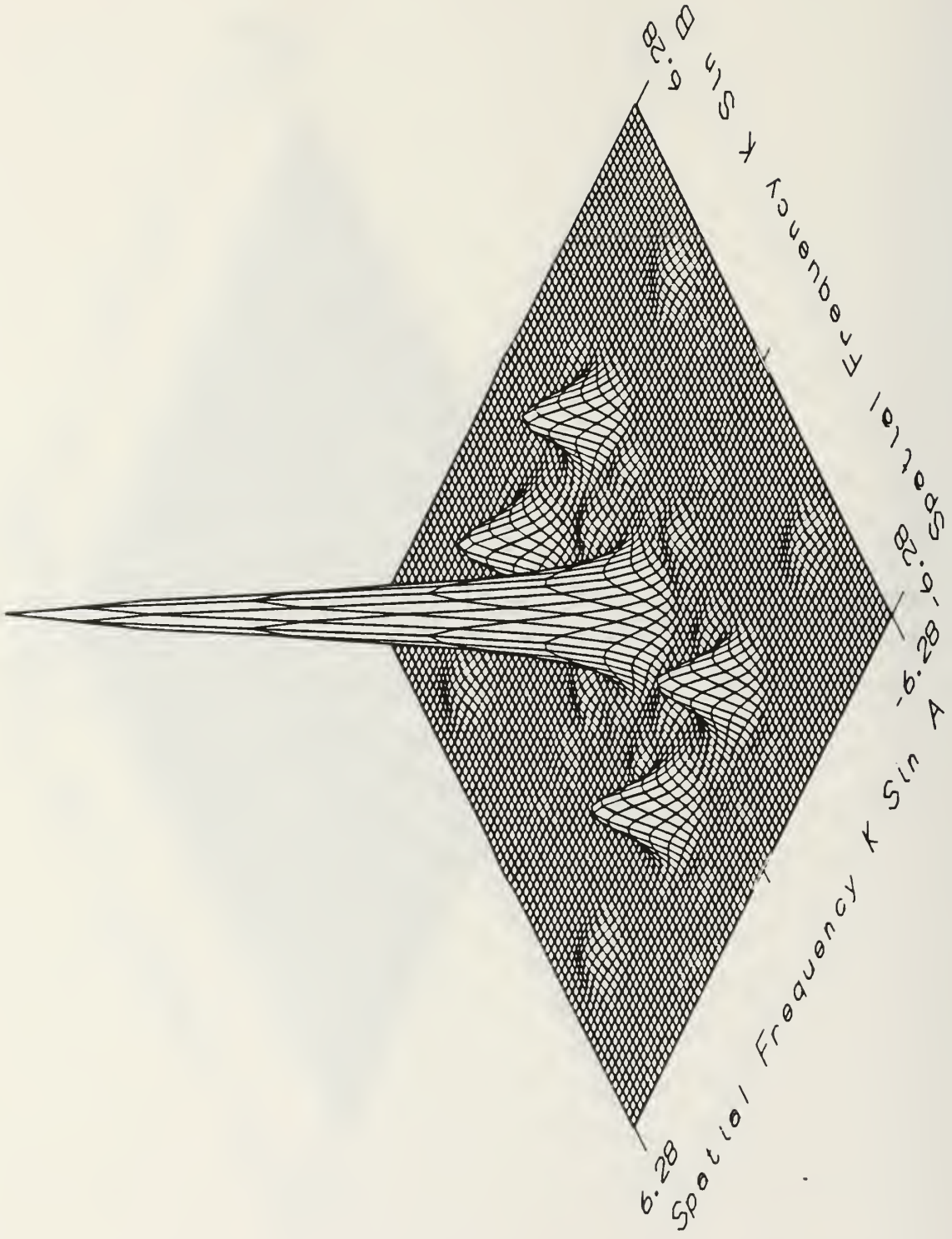

Figure B-22 Diffraction Pattern for Sequency 24

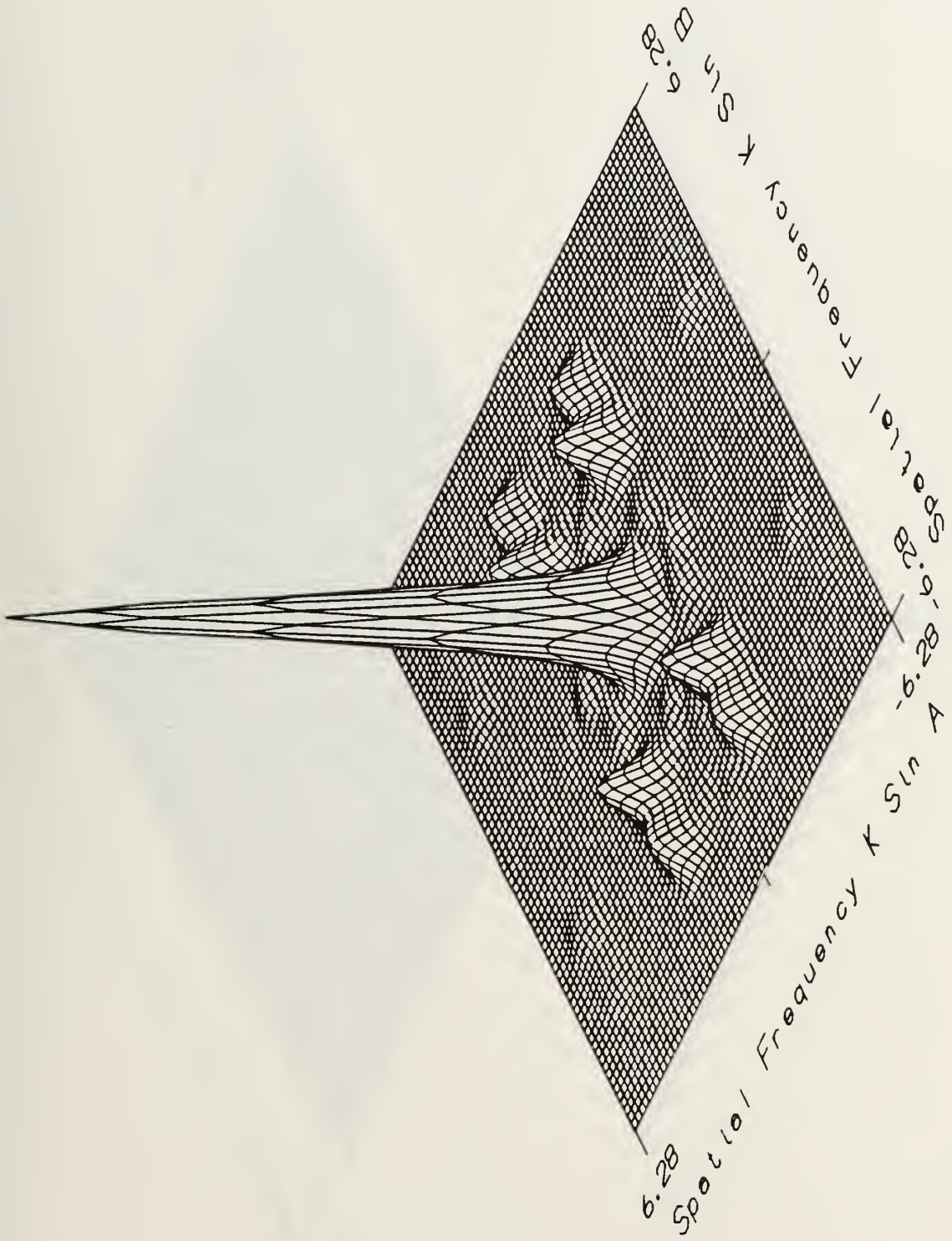

Figure B-23 Diffraction Pattern for Sequency 25

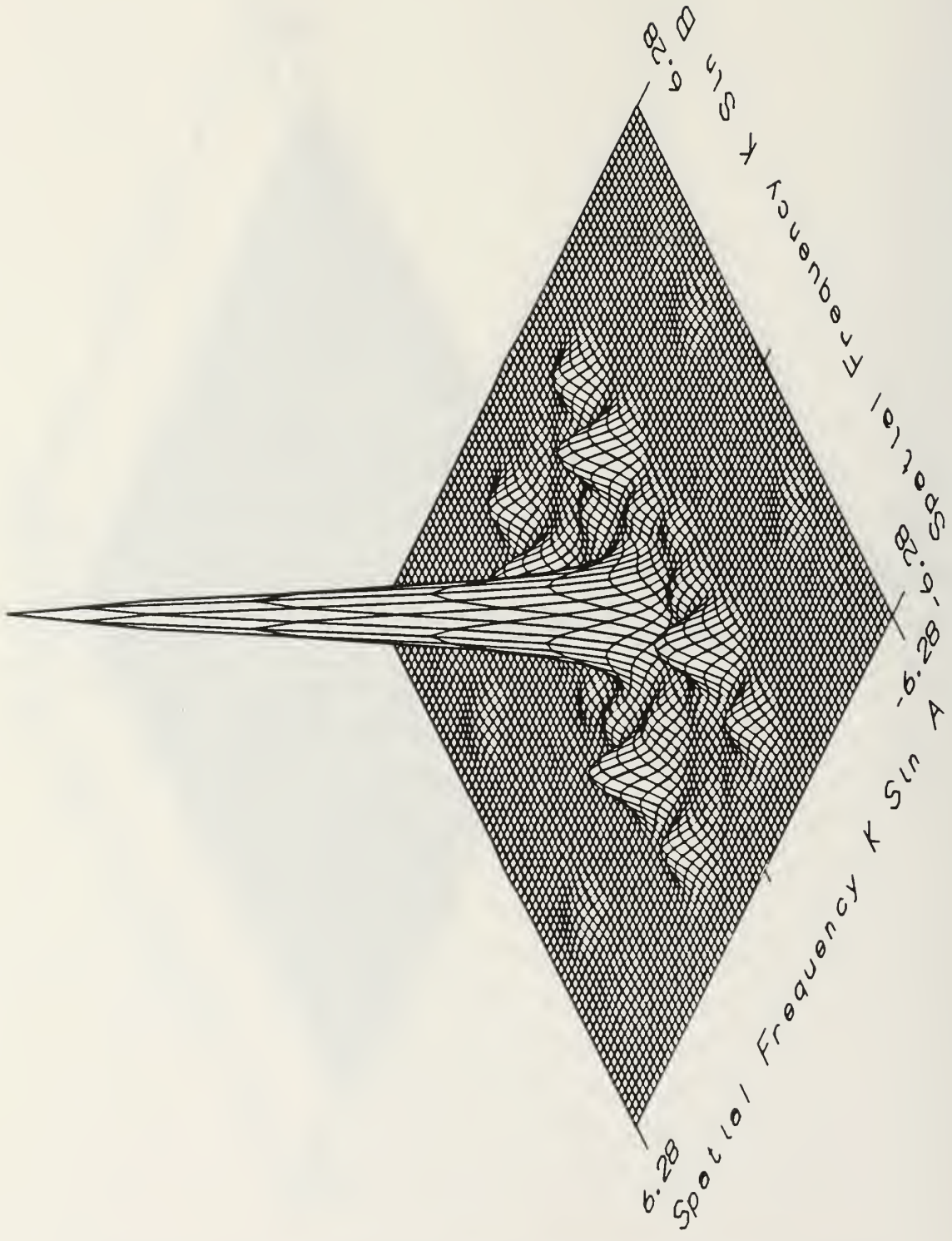

Figure B-24 Diffraction Pattern for Sequency 26

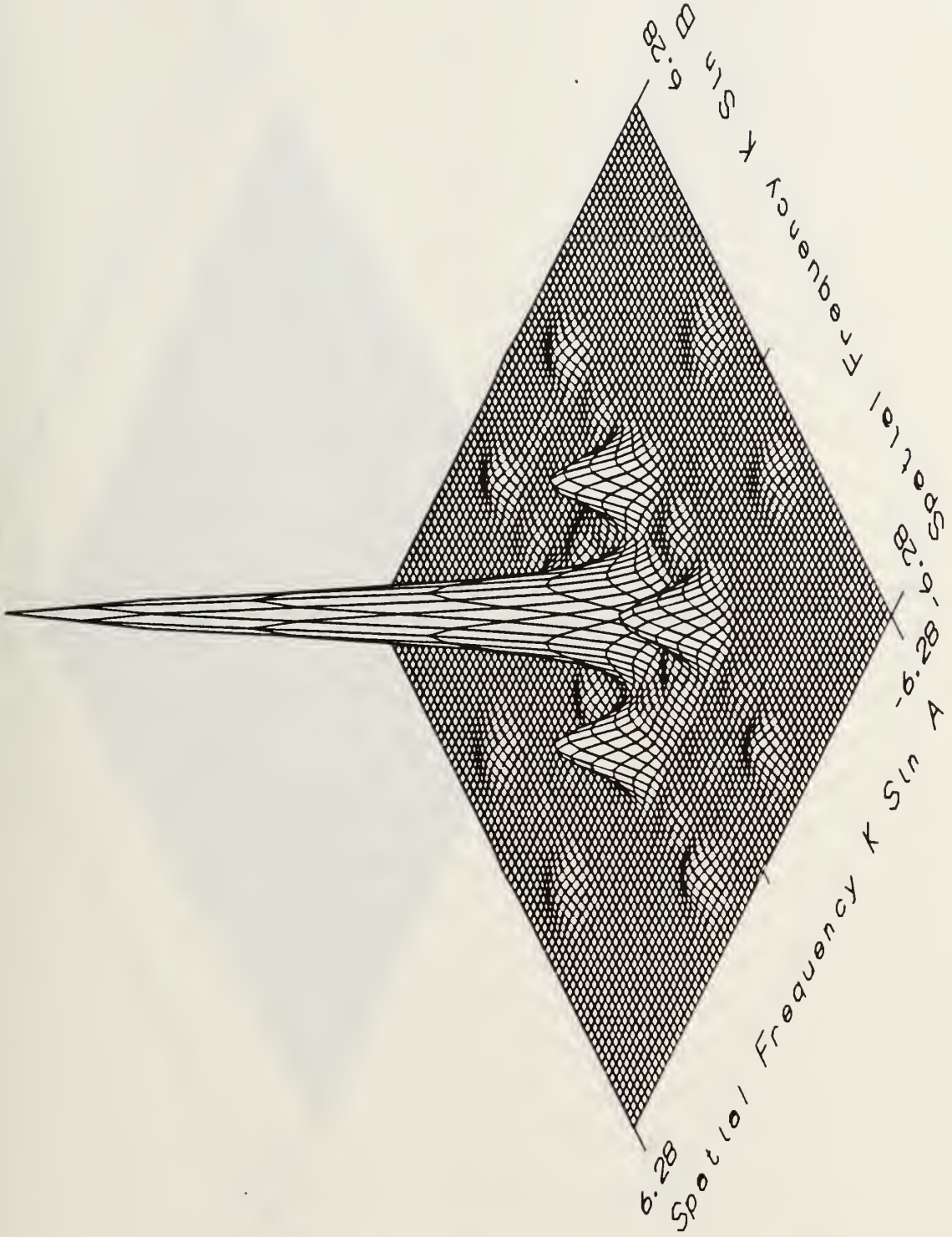

Figure B-25 Diffraction Pattern for Sequency 27

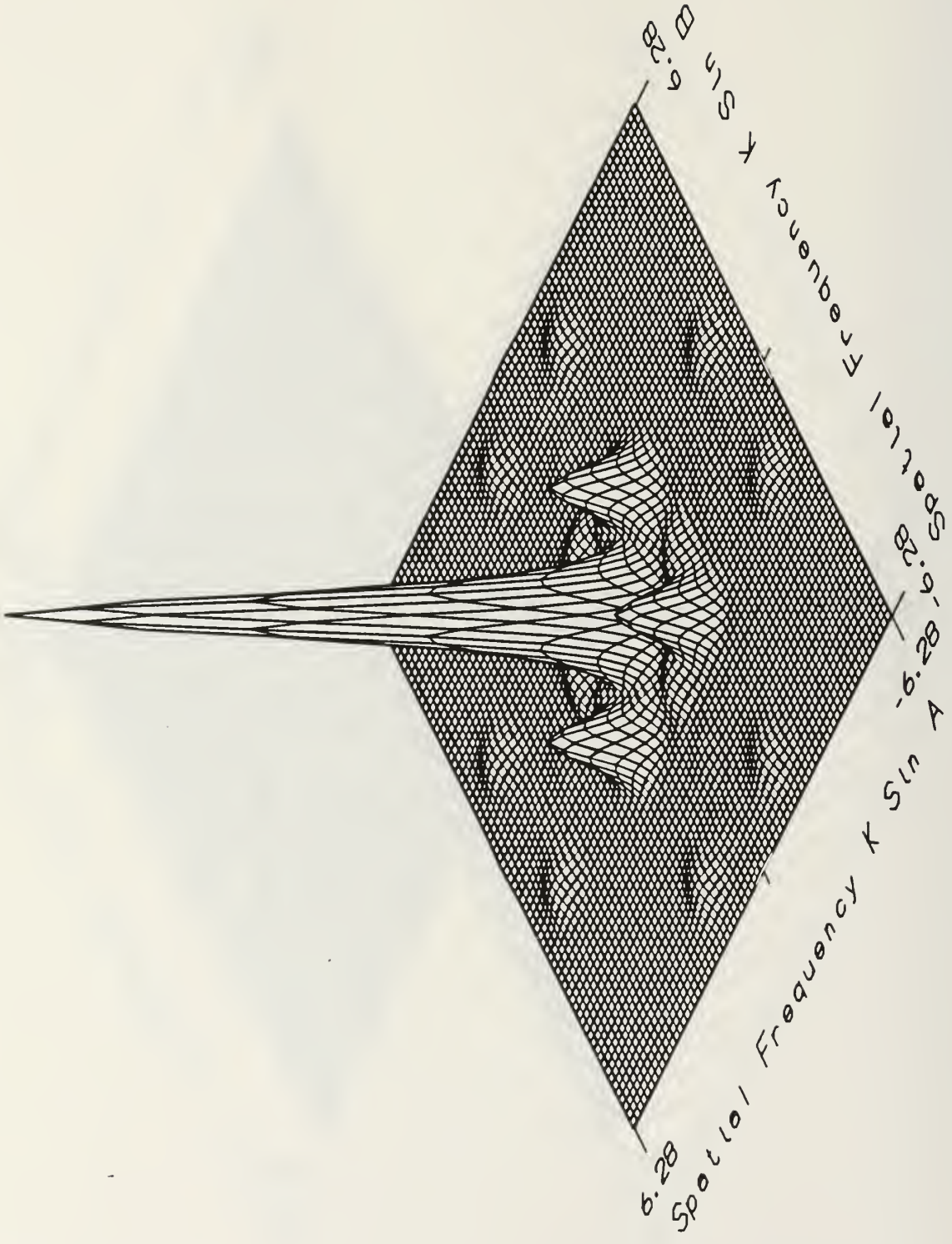

Figure B-26 Diffraction Pattern for Sequency 28

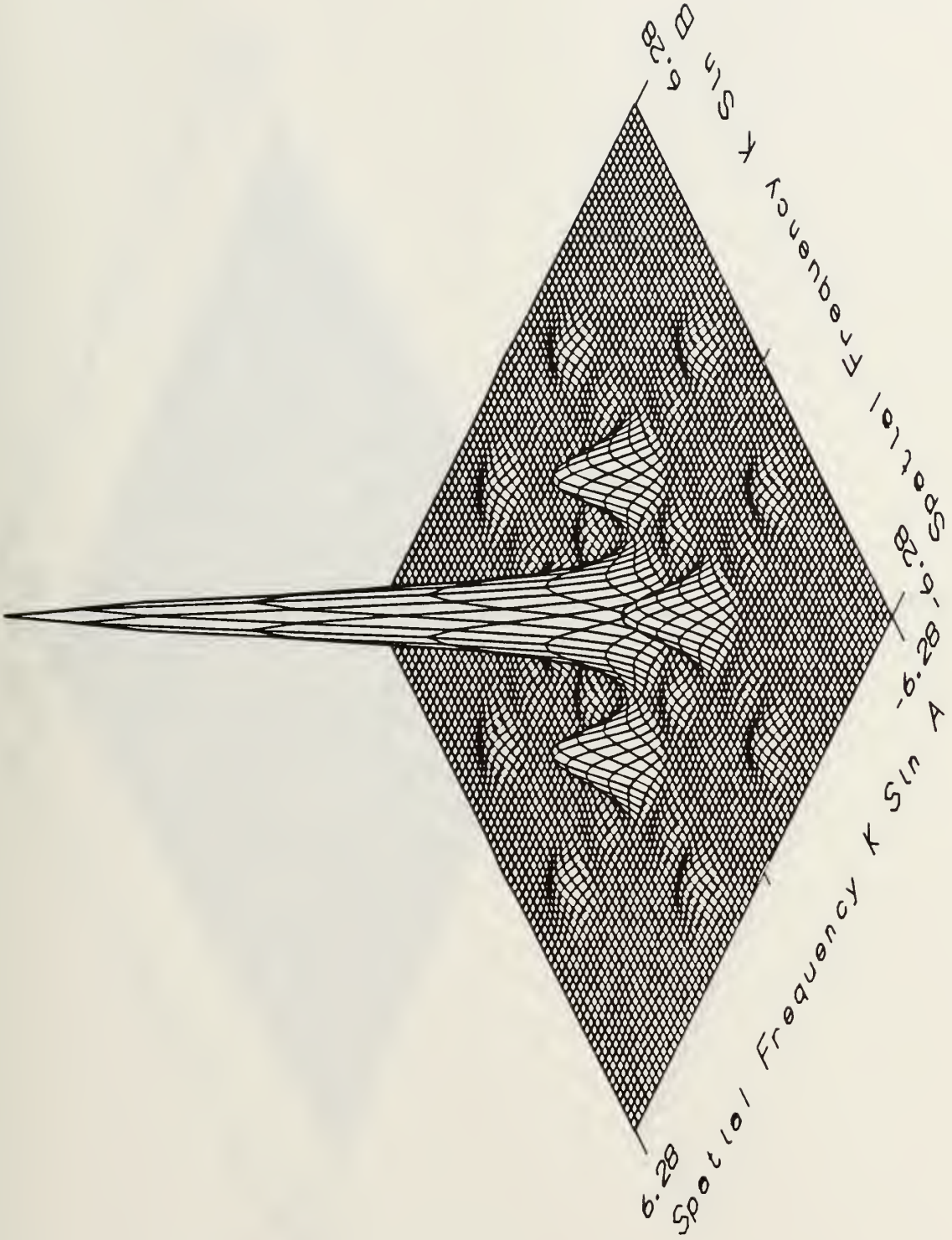

Figure B-27 Diffraction Pattern for Sequency 36

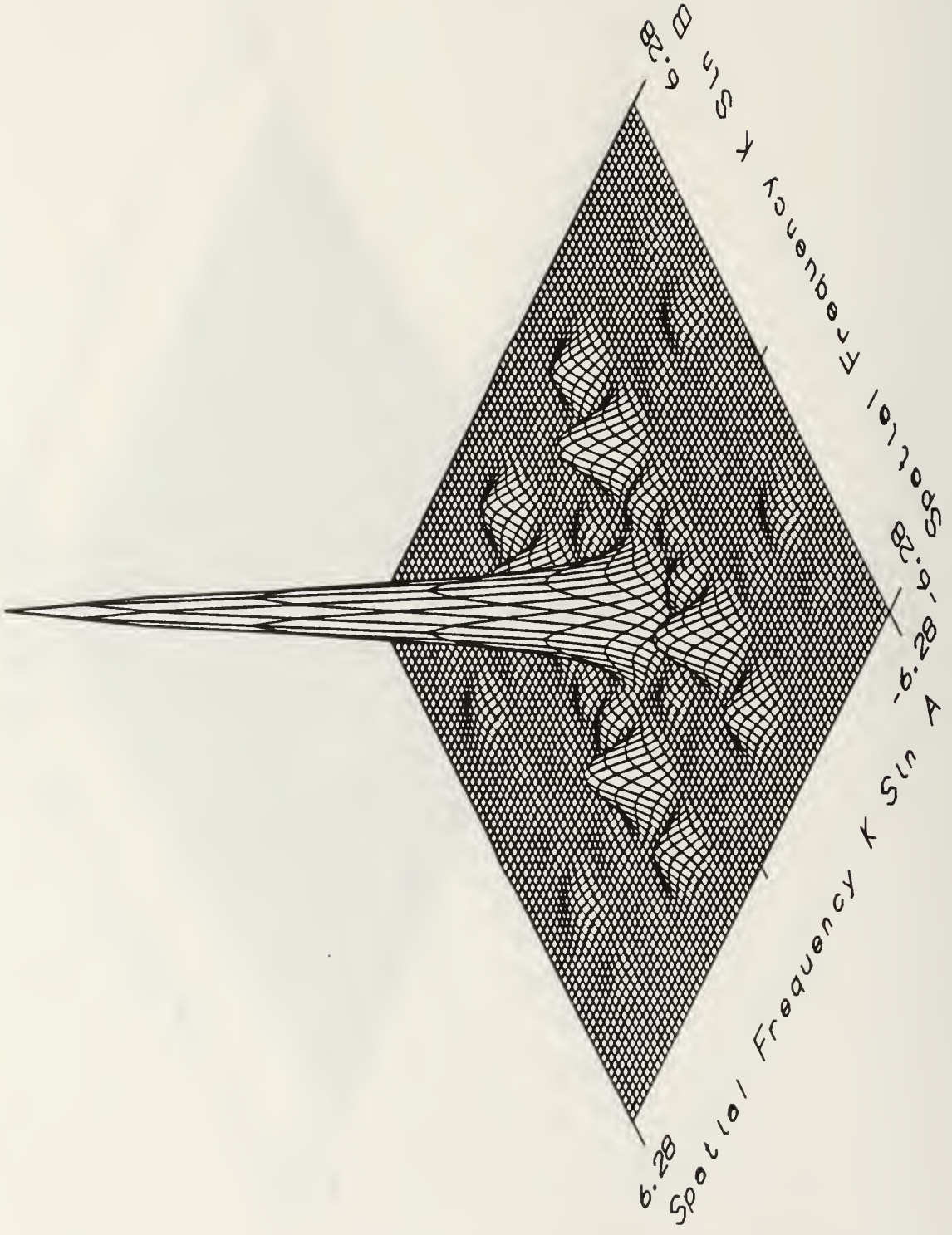

Figure B-28 Diffraction Pattern for Sequency 37

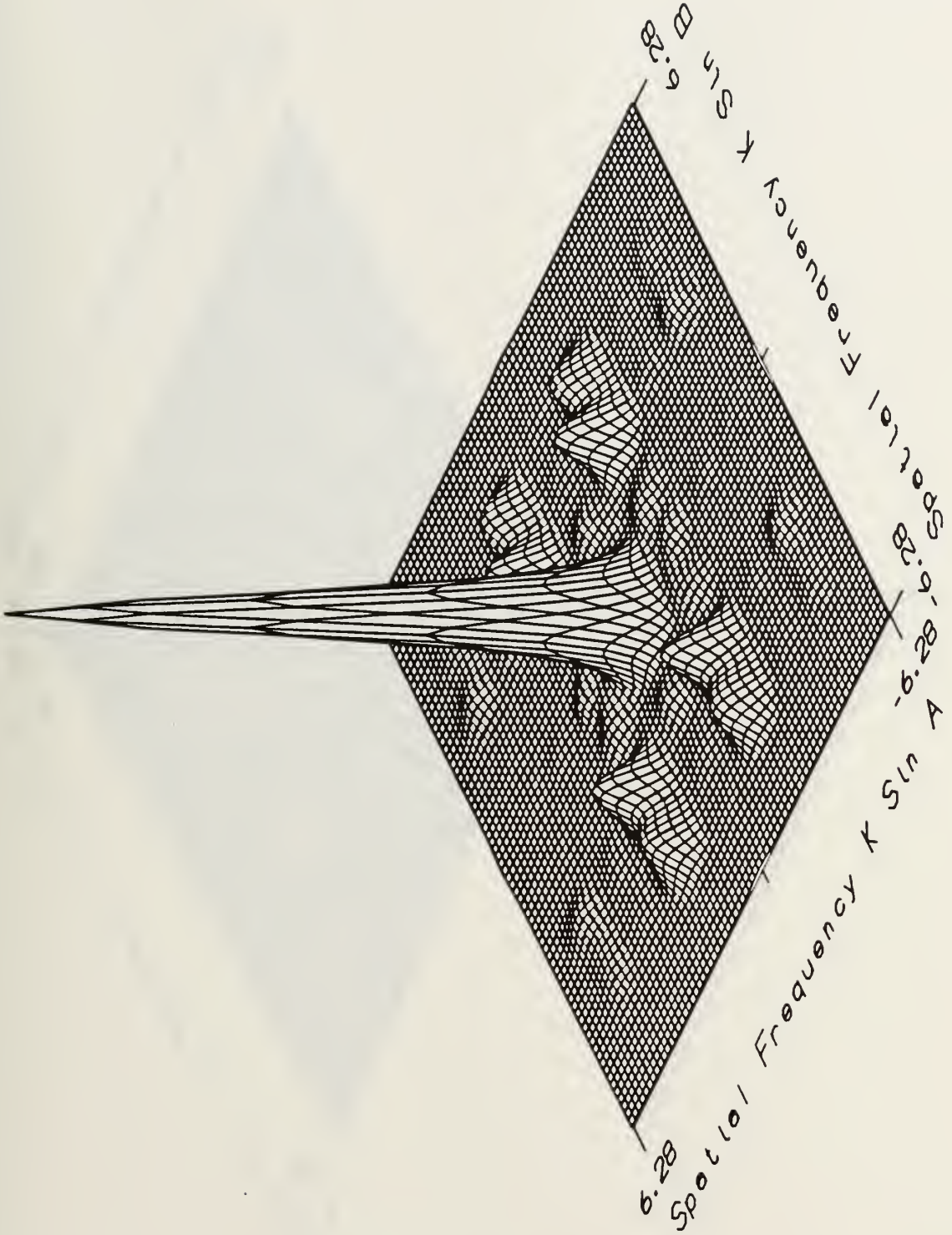

Figure B-29 Diffraction Pattern for Sequency 38

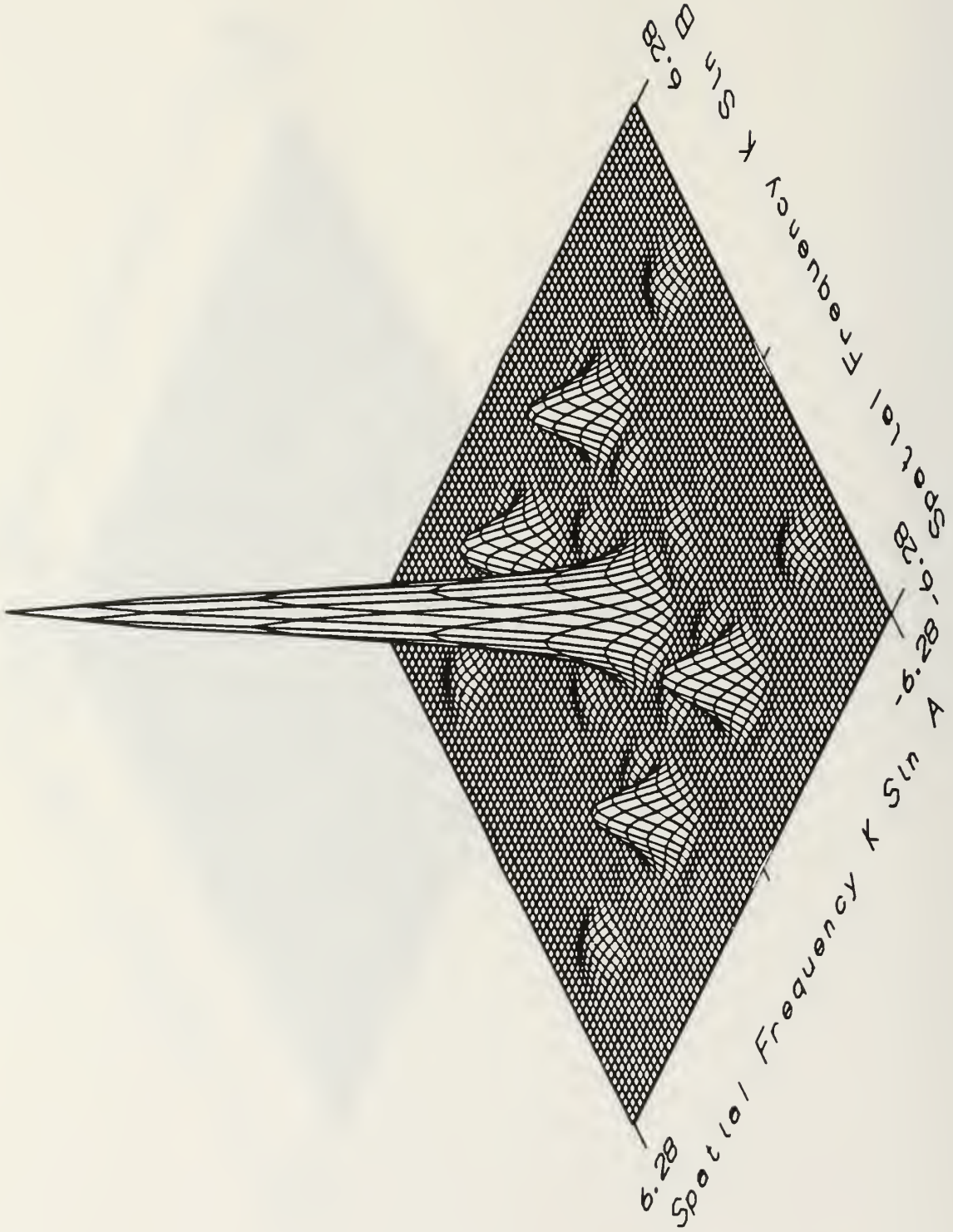

Figure B-30 Diffraction Pattern for Sequency 39

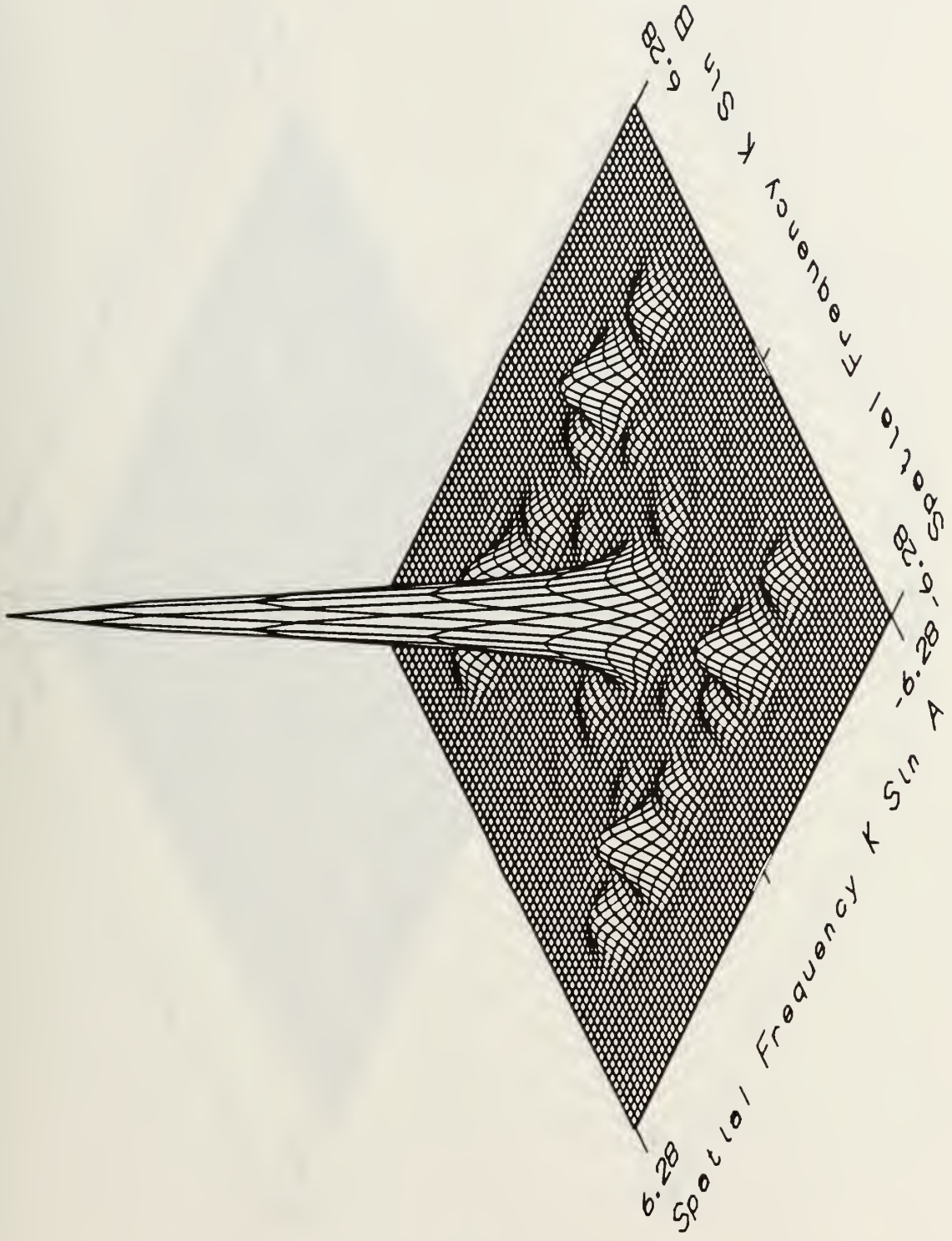

Figure B-31 Diffraction Pattern for Sequency 40

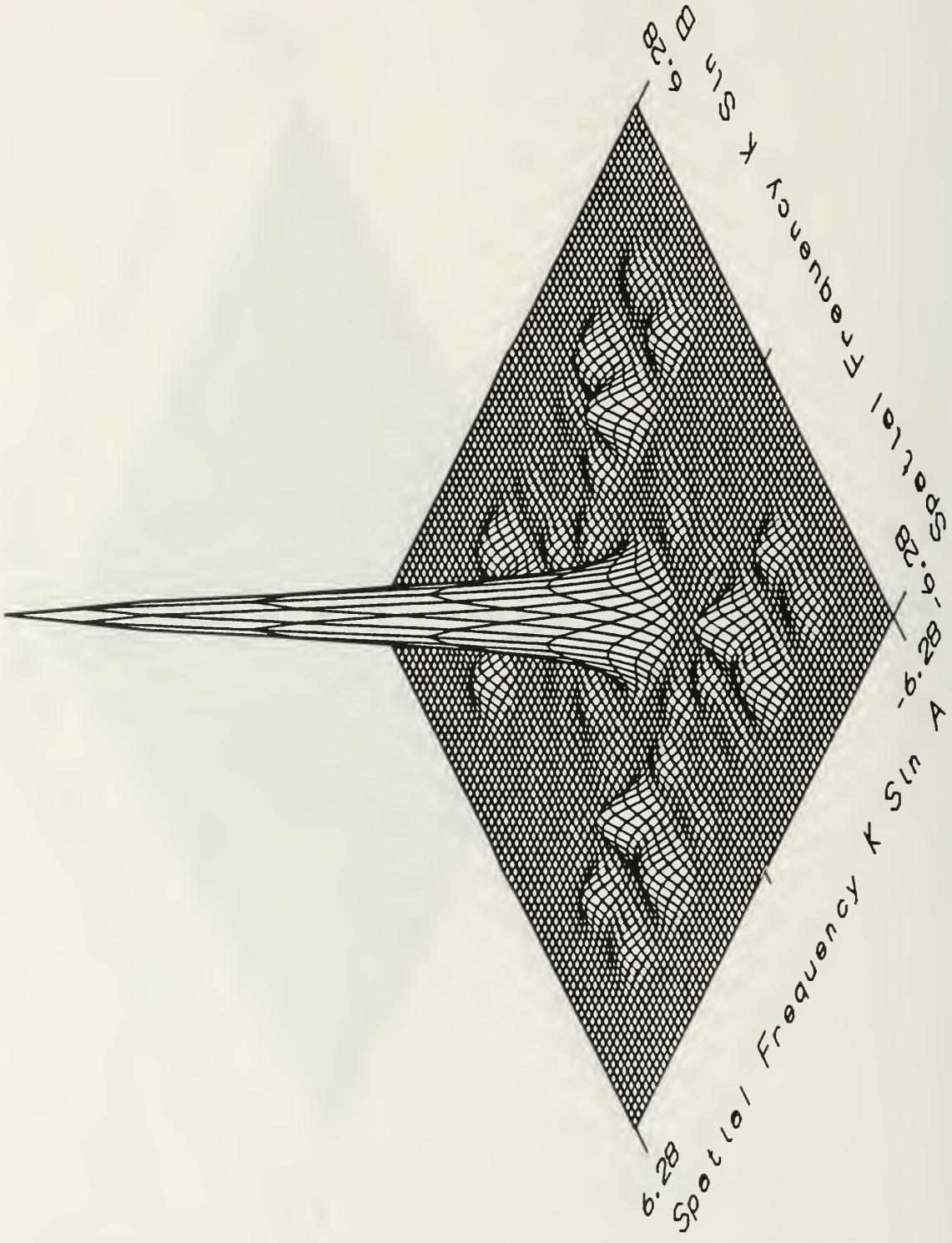

Figure B-32 Diffraction Pattern for Sequency 41

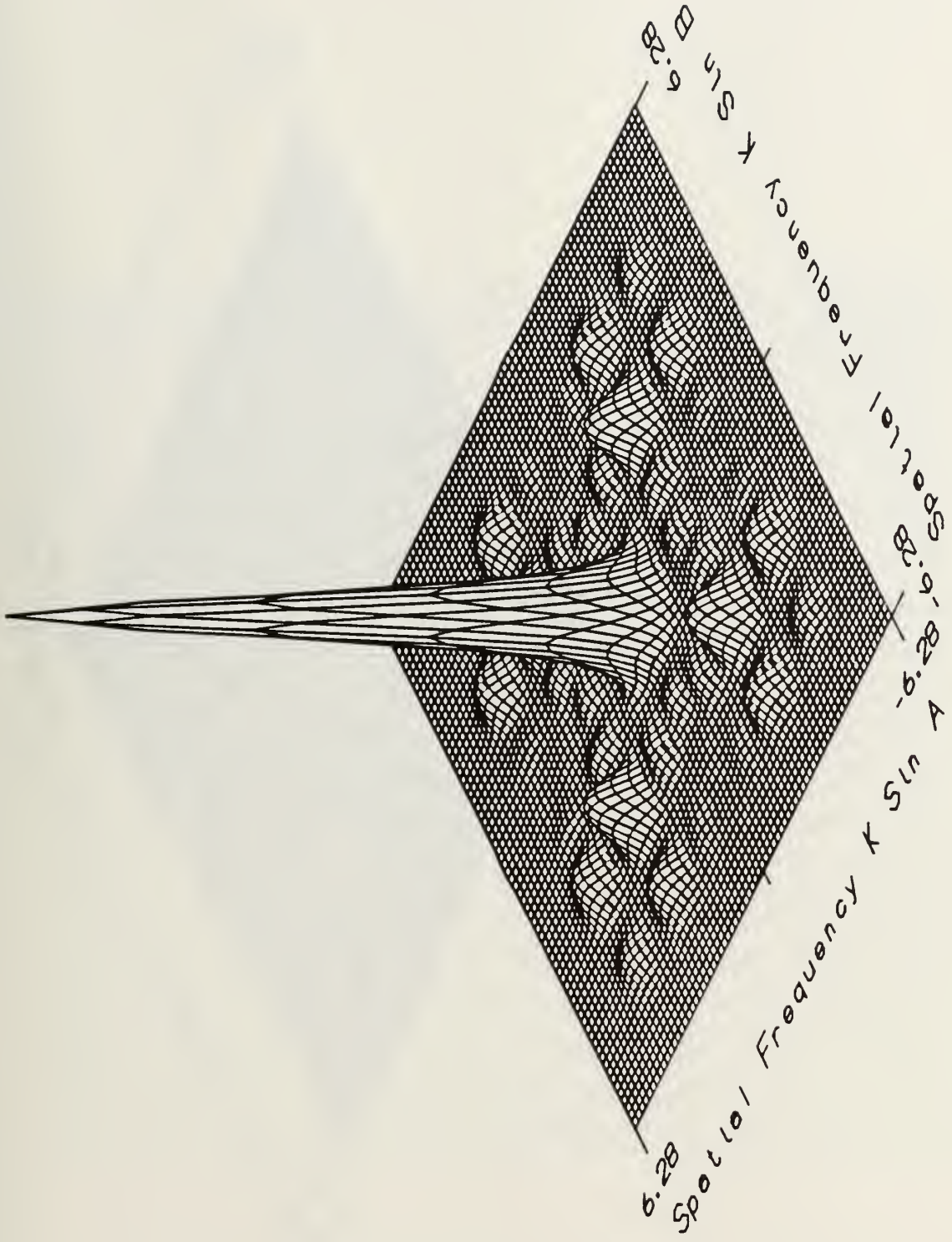

Figure B-33 Diffraction Pattern for Sequency 42

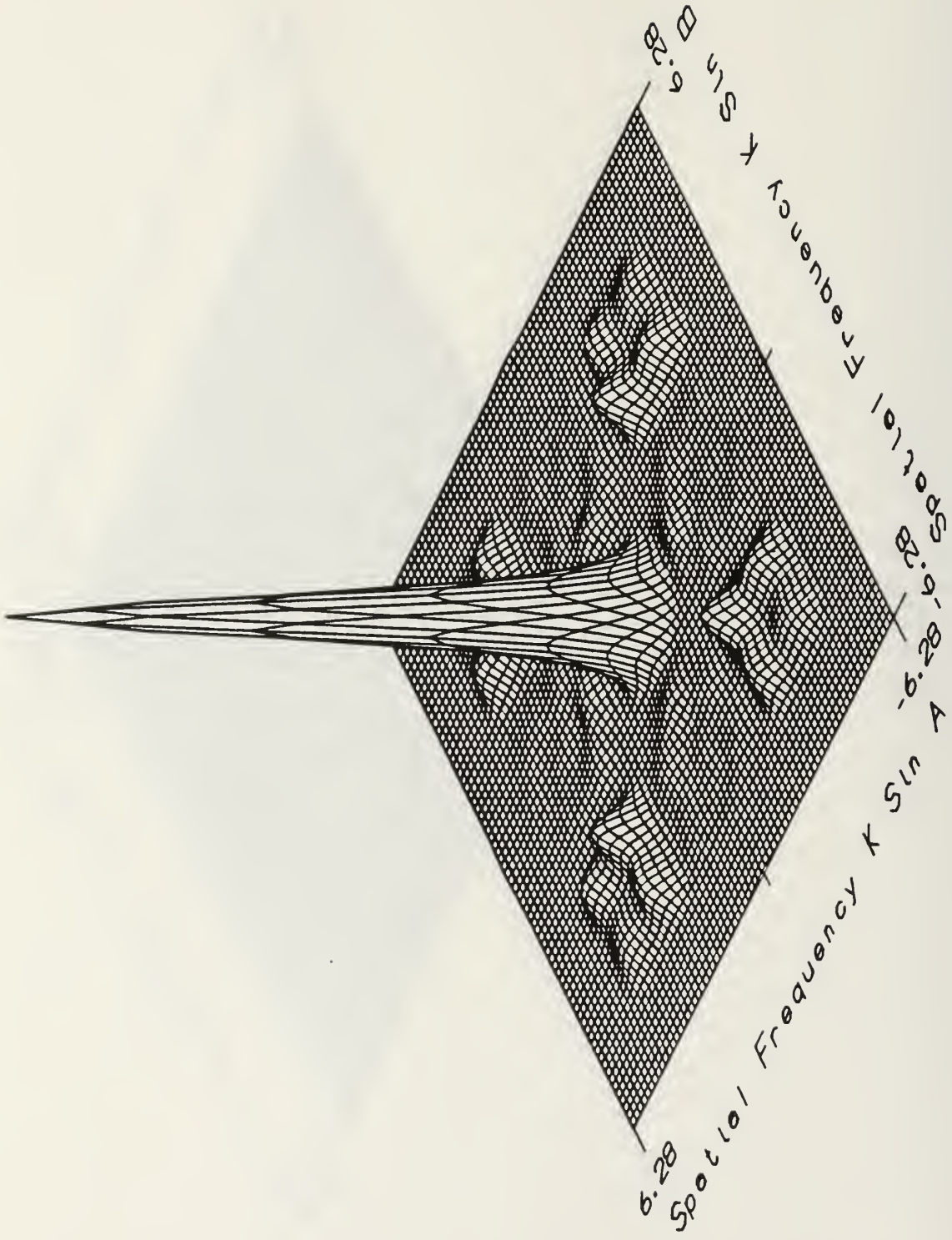

Figure B-34 Diffraction Pattern for Sequency 54

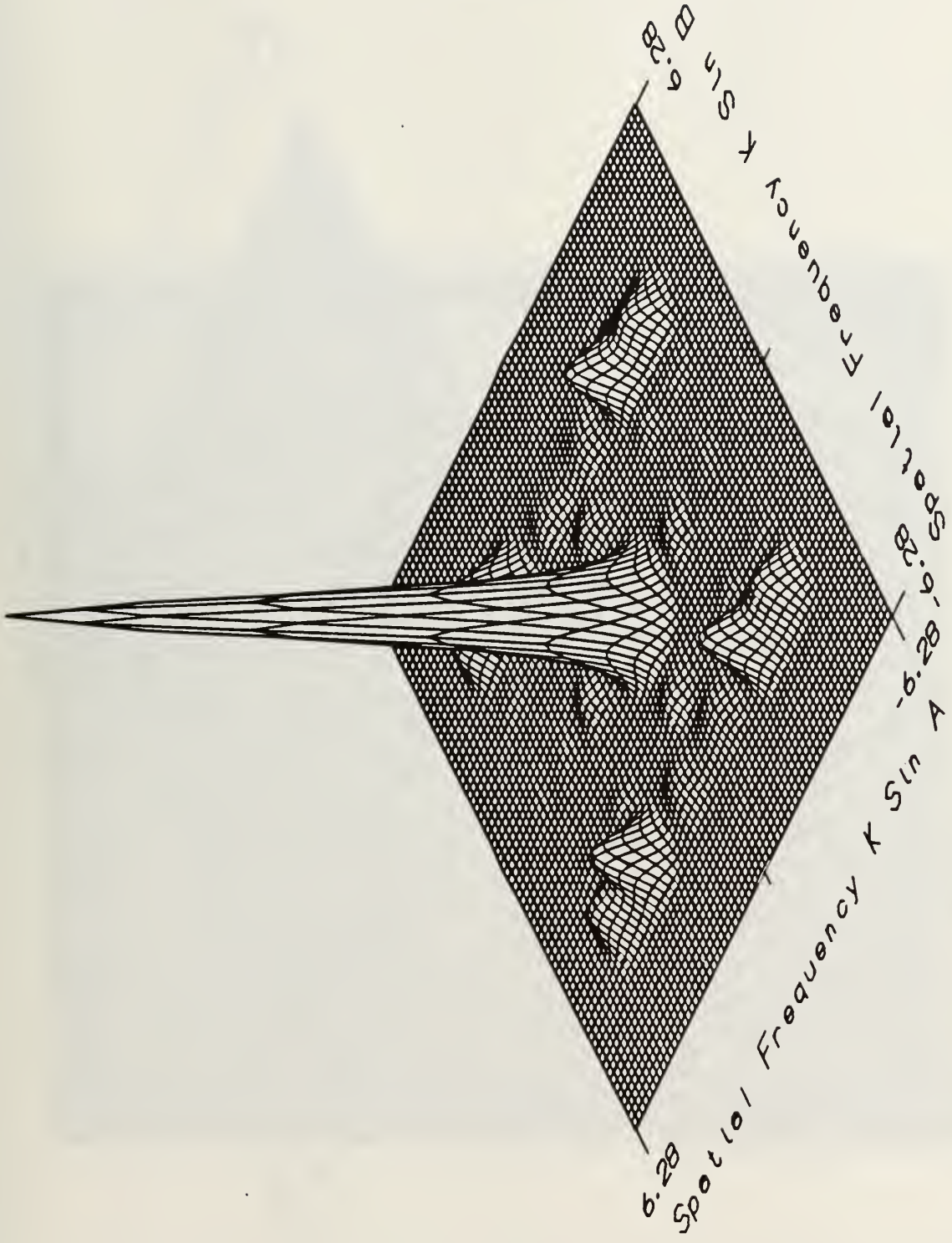

Figure B-35 Diffraction Pattern for Sequency 55

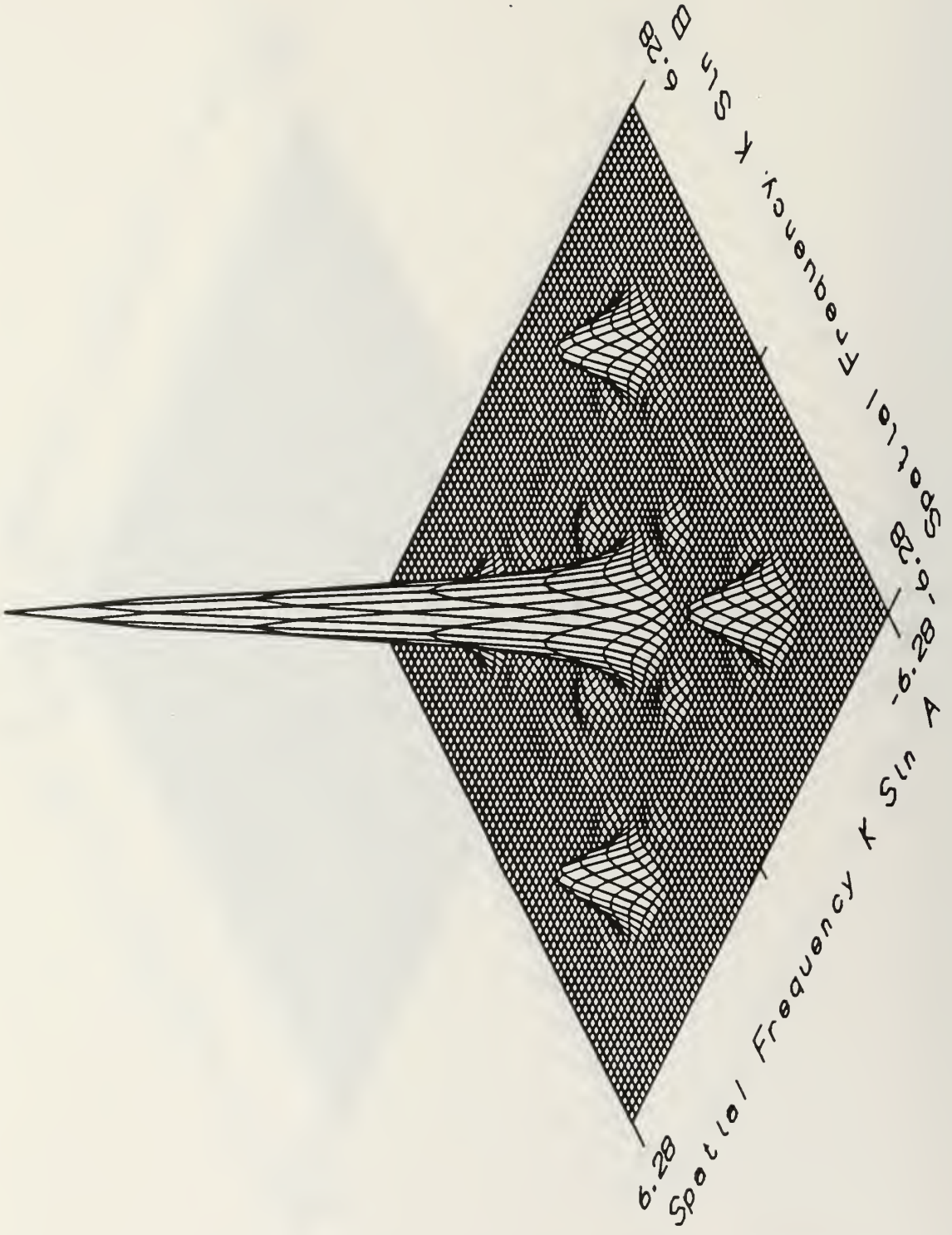

Figure B-36 Diffraction Pattern for Sequency 56

APPENDIX C

Figure C-1 Cutter512V

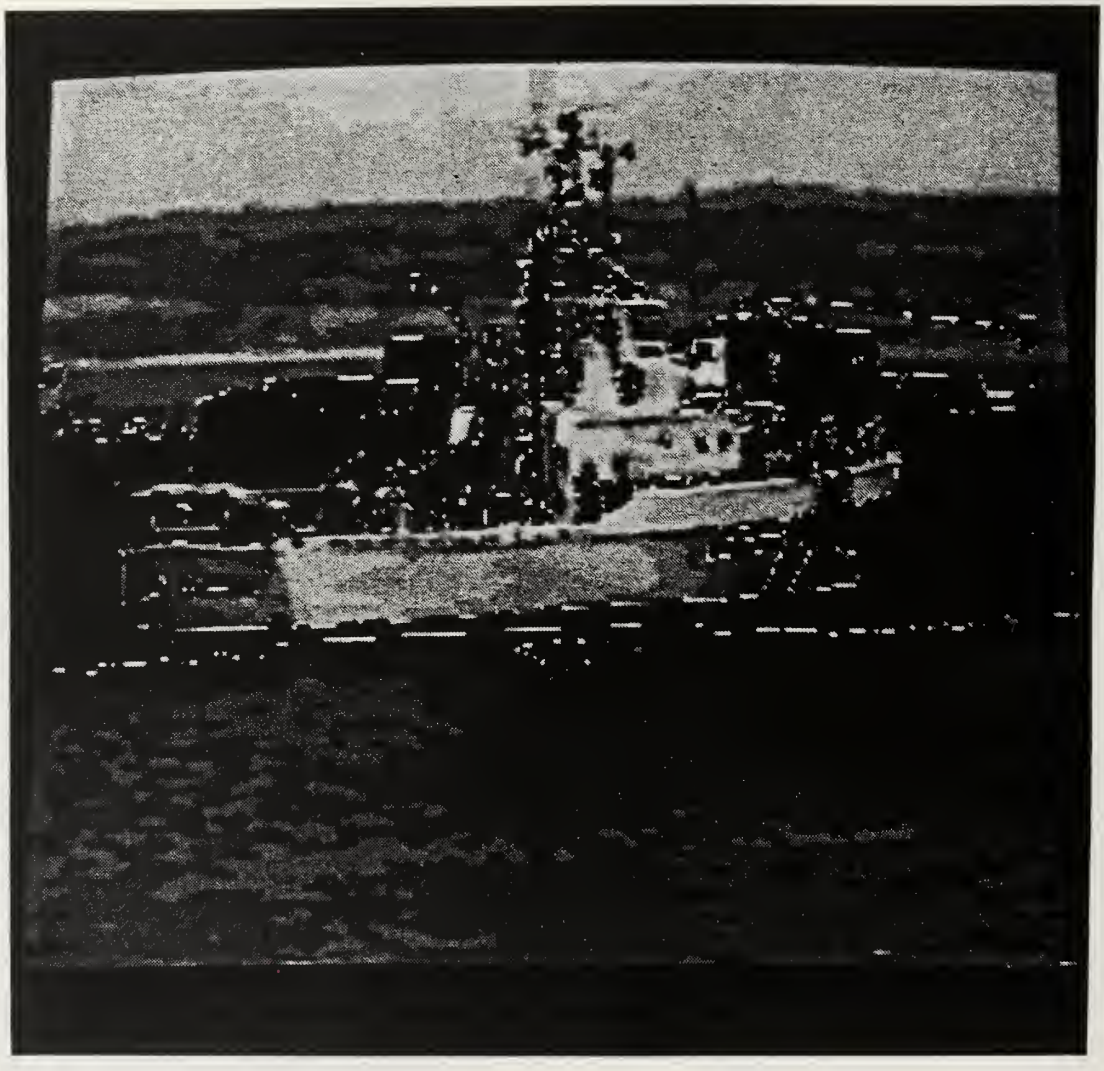

Figure C-2 Cutter256V

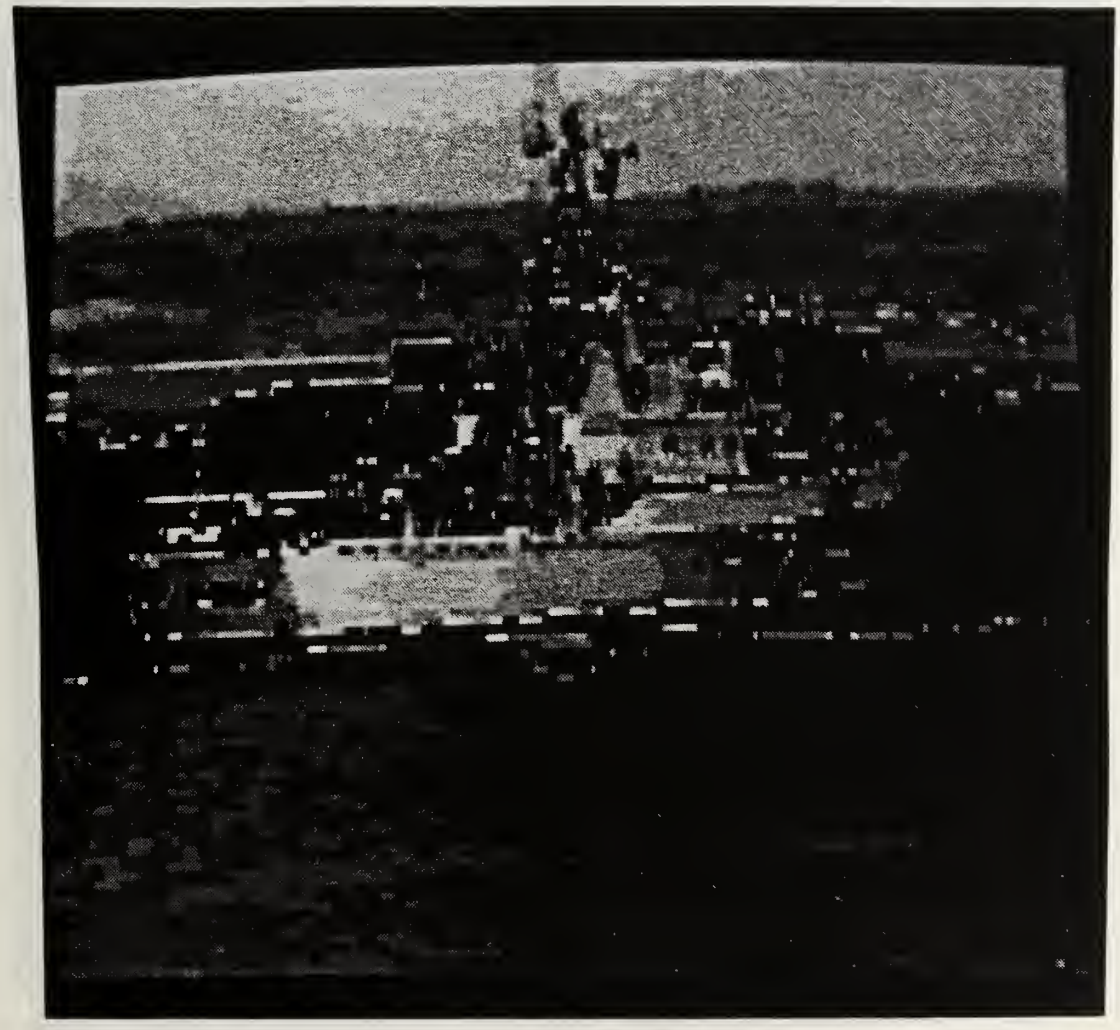

Figure C-3 Cutter128V

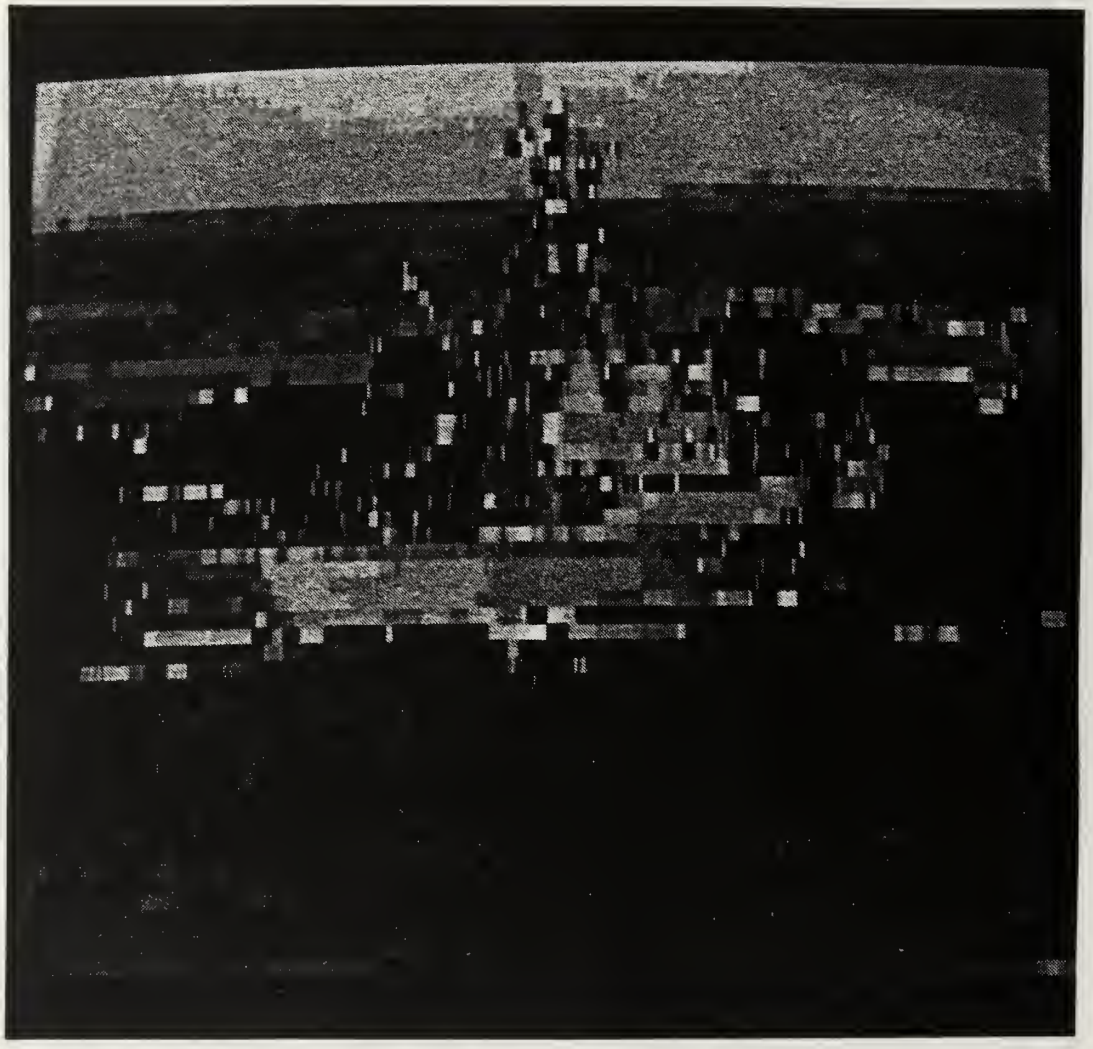

Figure C-4 Cutter64V

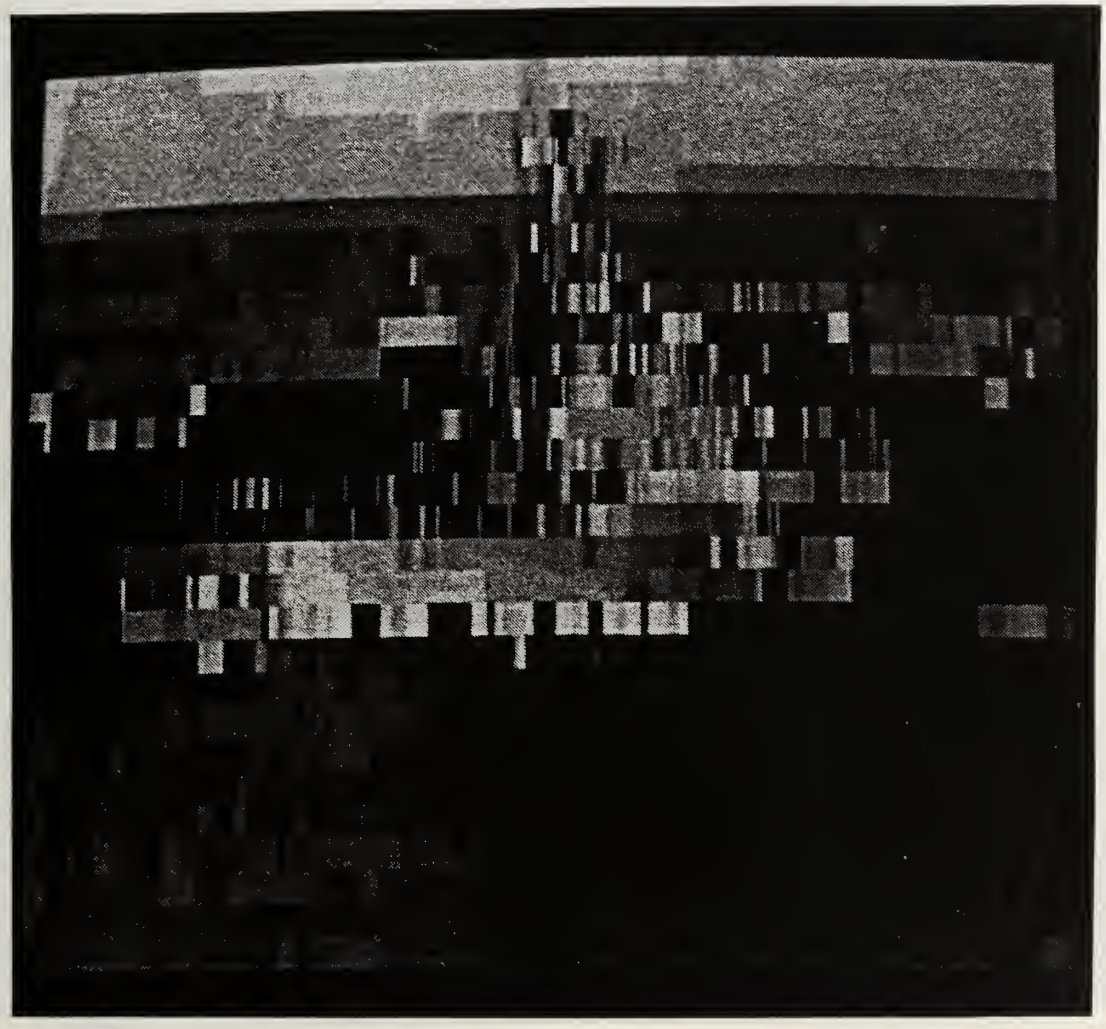

Figure C-5 Cutter32V

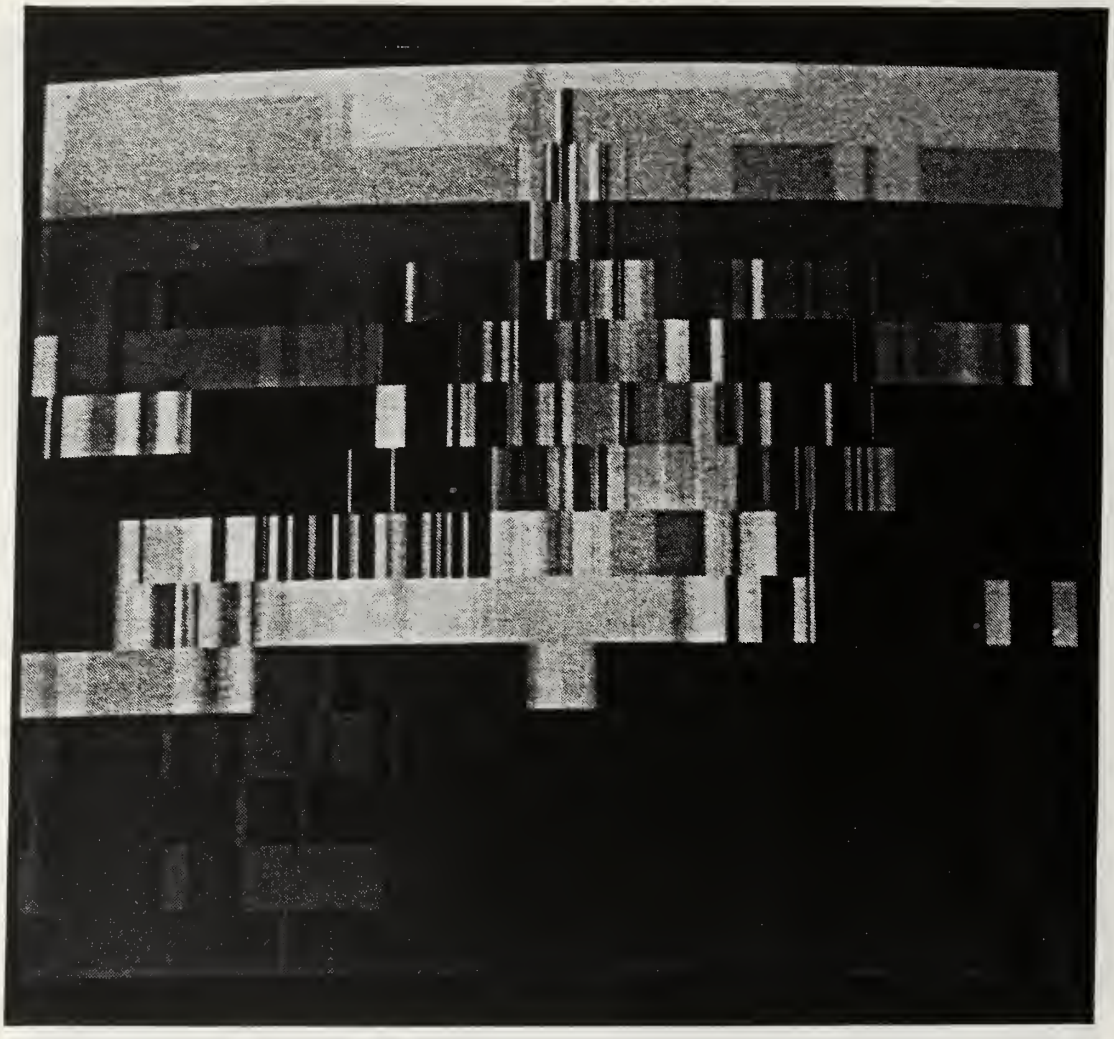

Figure C-6 Cutter16

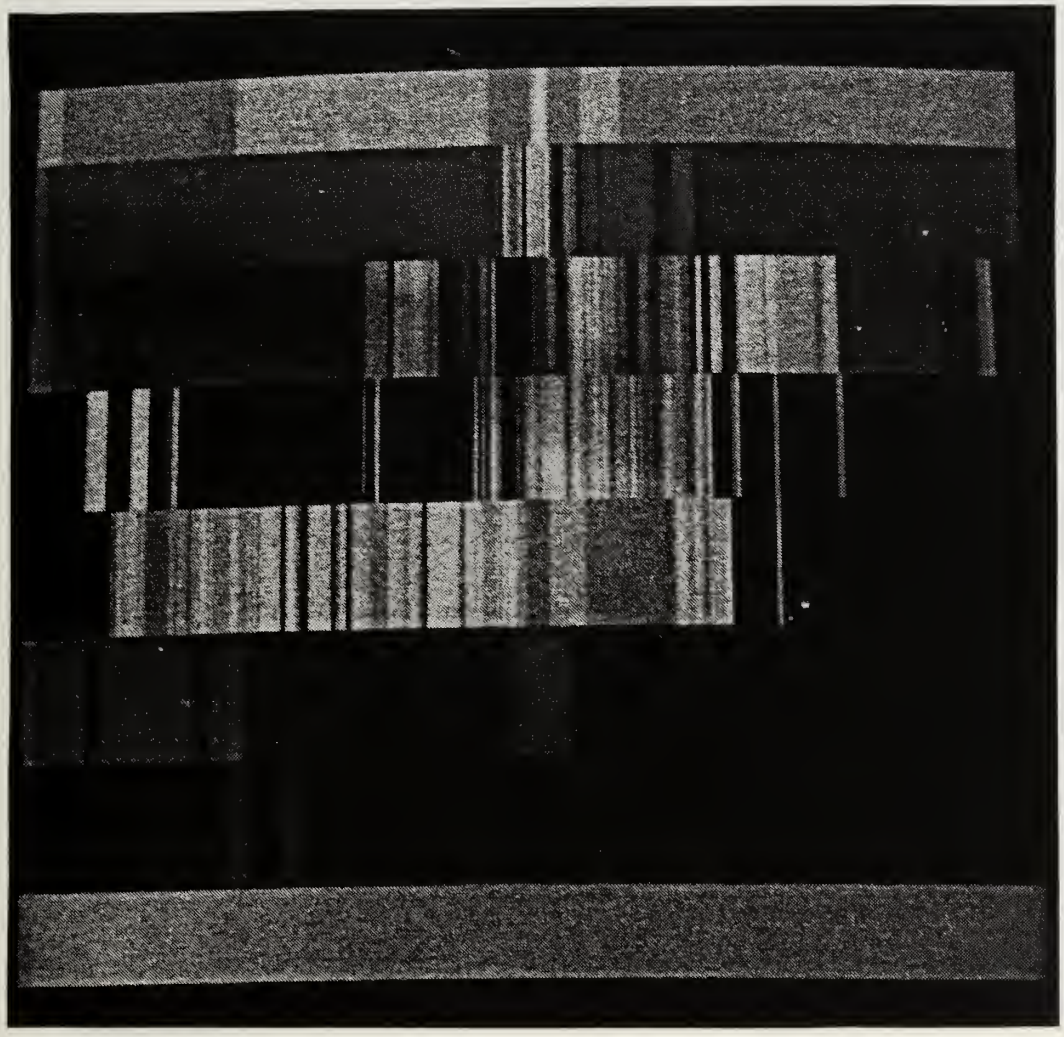

Figure C-7 Cutter8V

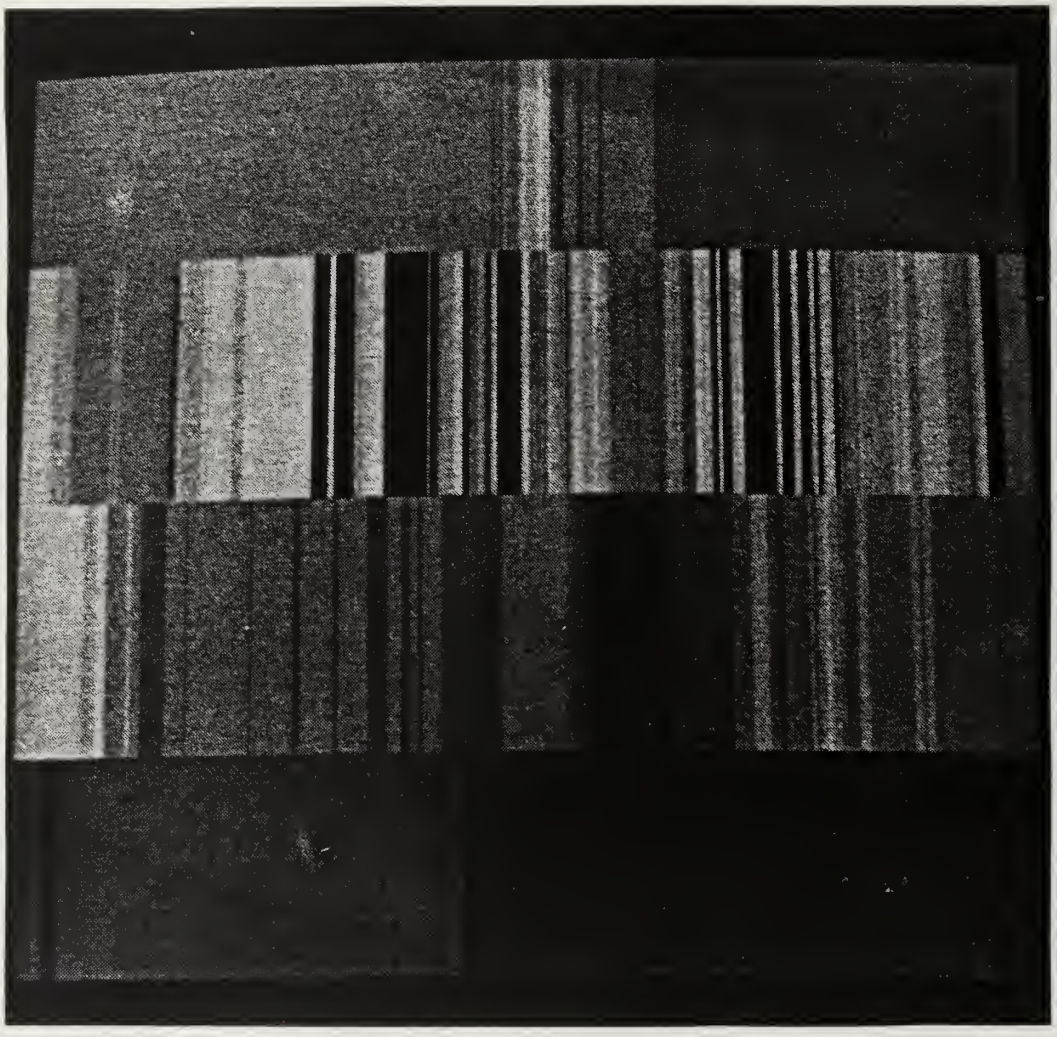

Figure C-8 Cutter4V

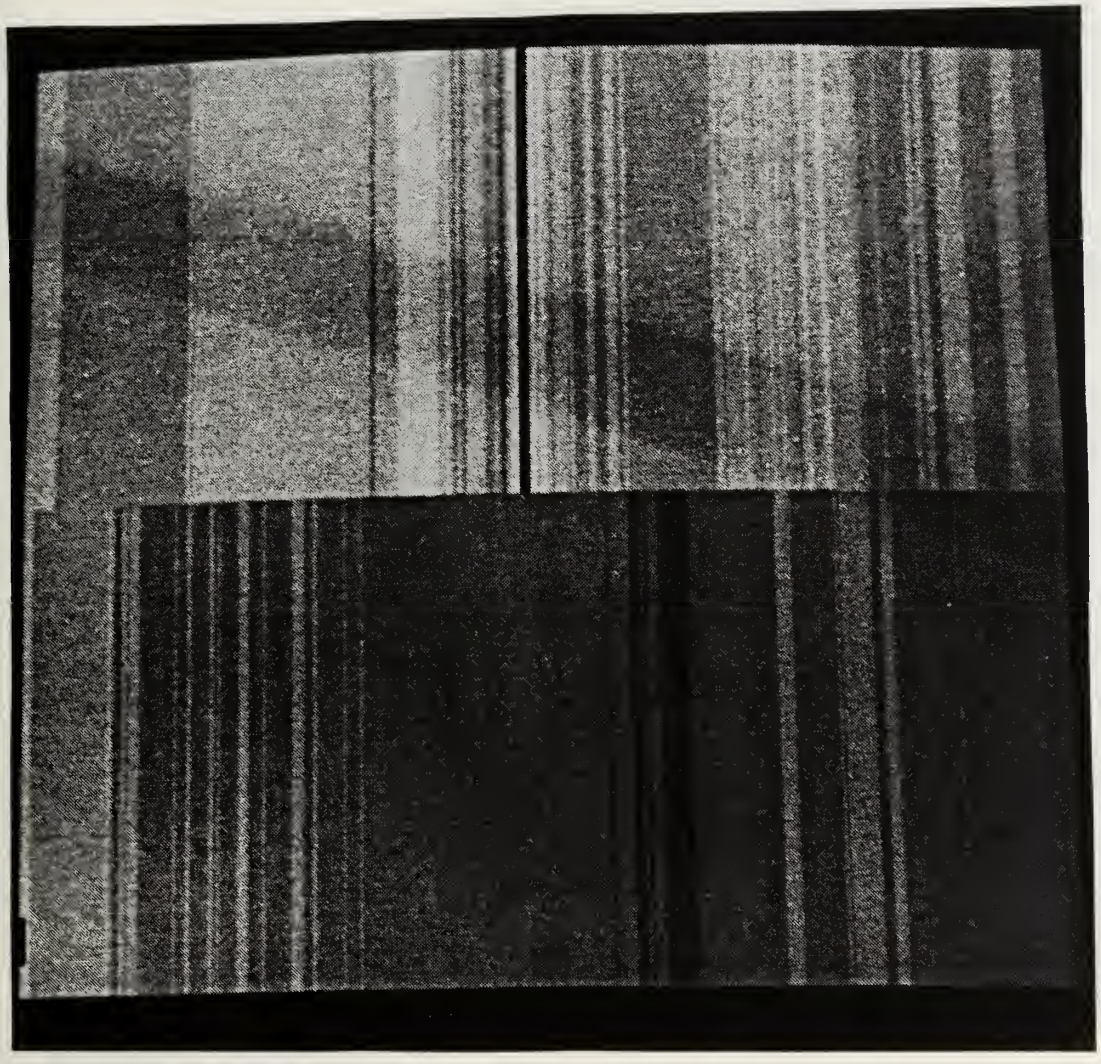

Figure C-9 Cutter2V

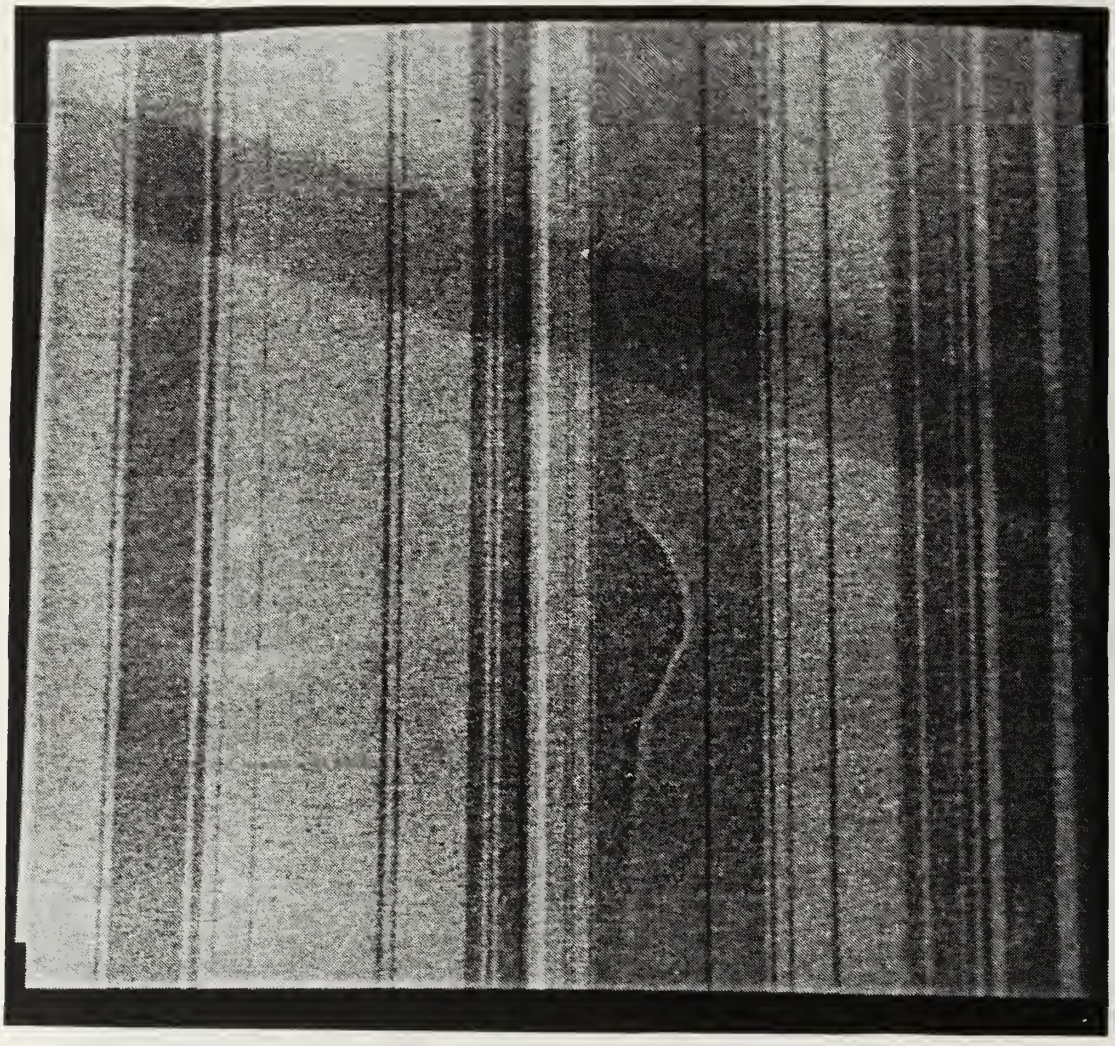

Figure C-10 Cutter1V

APPENDIX D

Figure D-1 Cutter512H

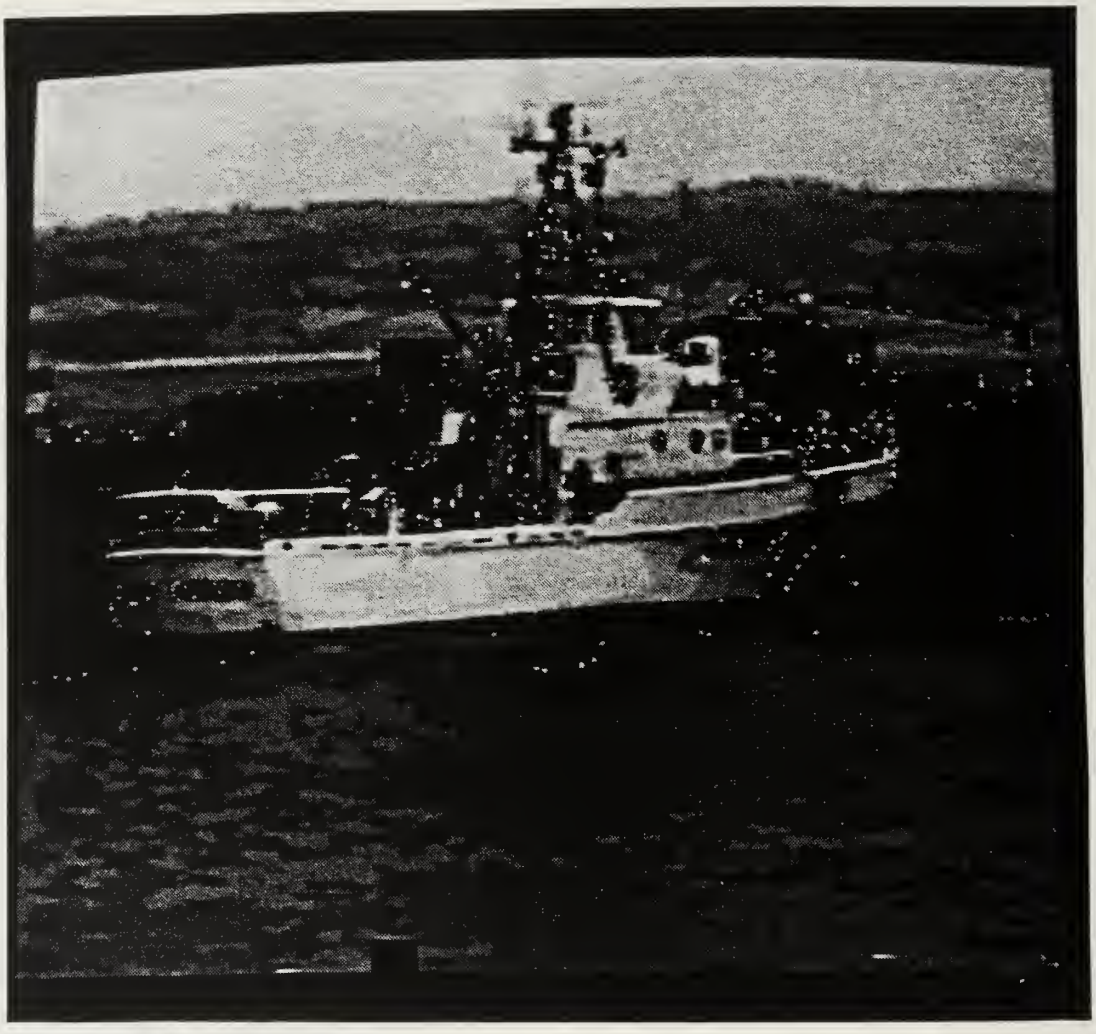

Figure D-2 Cutter256H

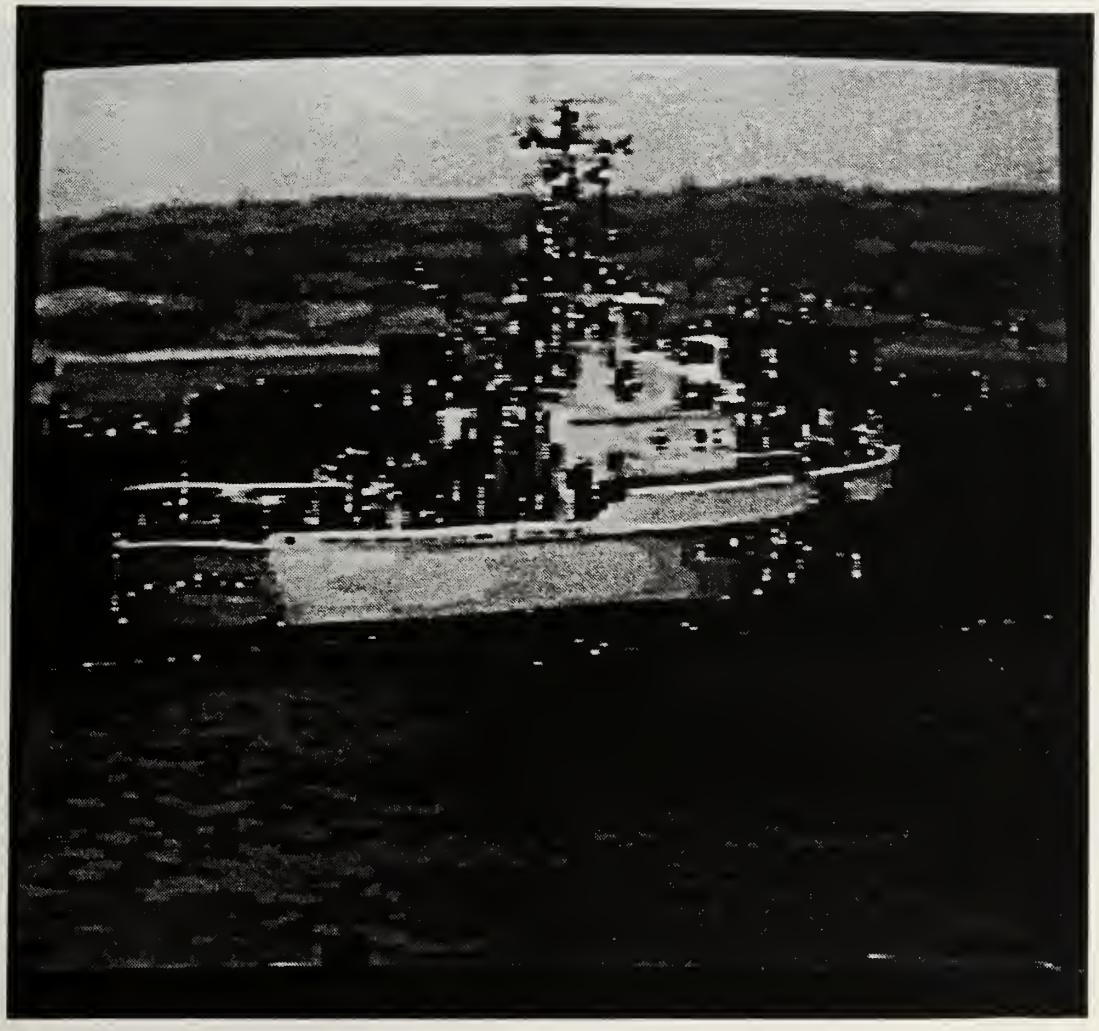

Figure D-3 Cutter128H

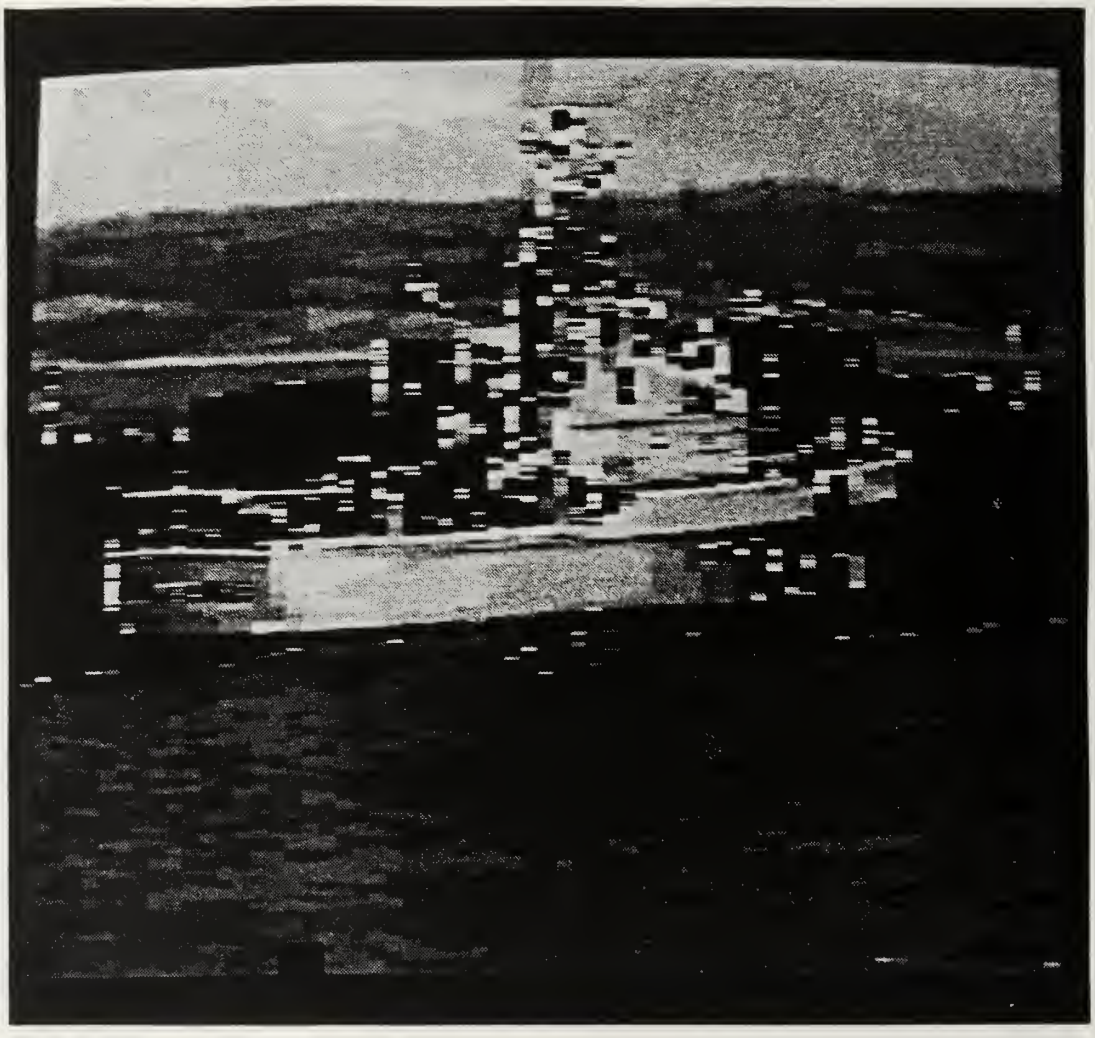

Figure D-4 Cutter64H

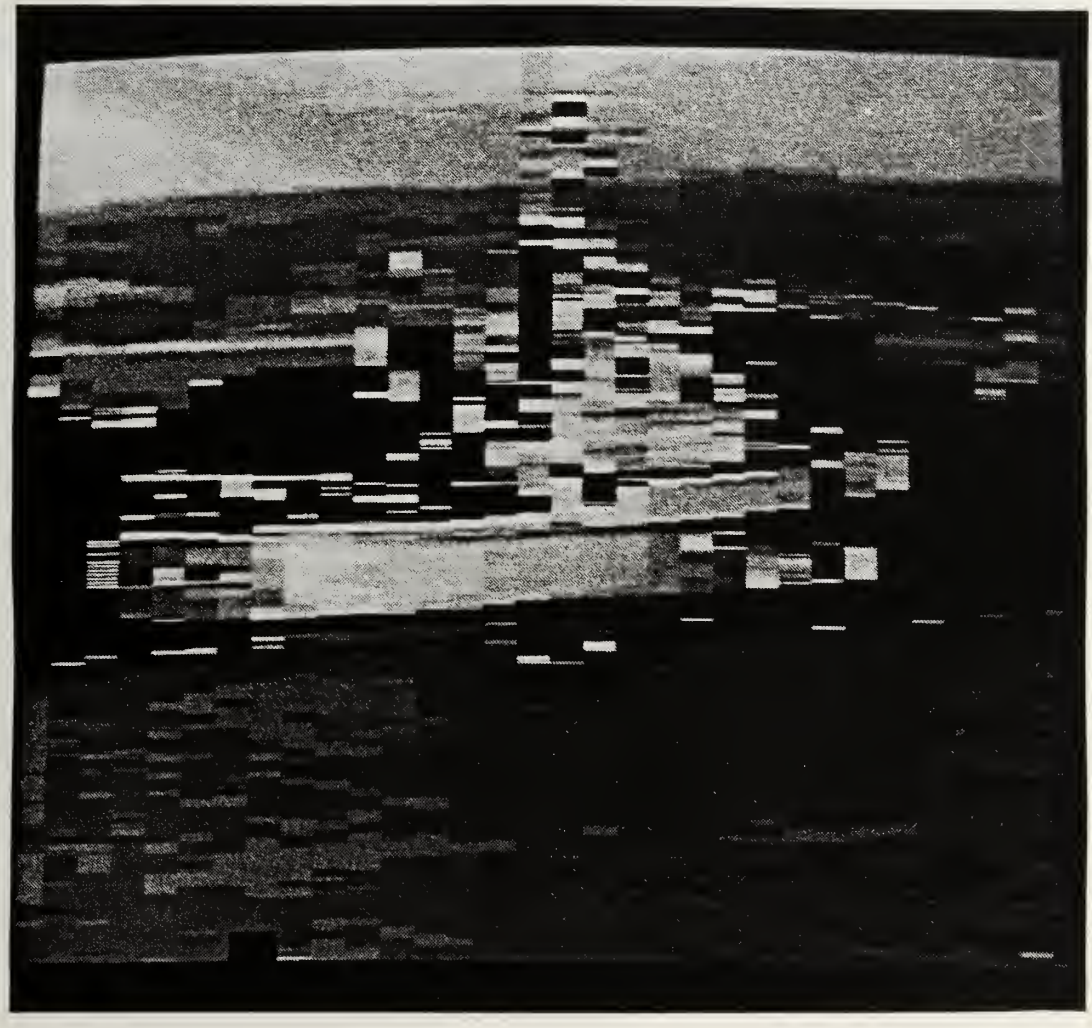

Figure D-5 Cutter32H

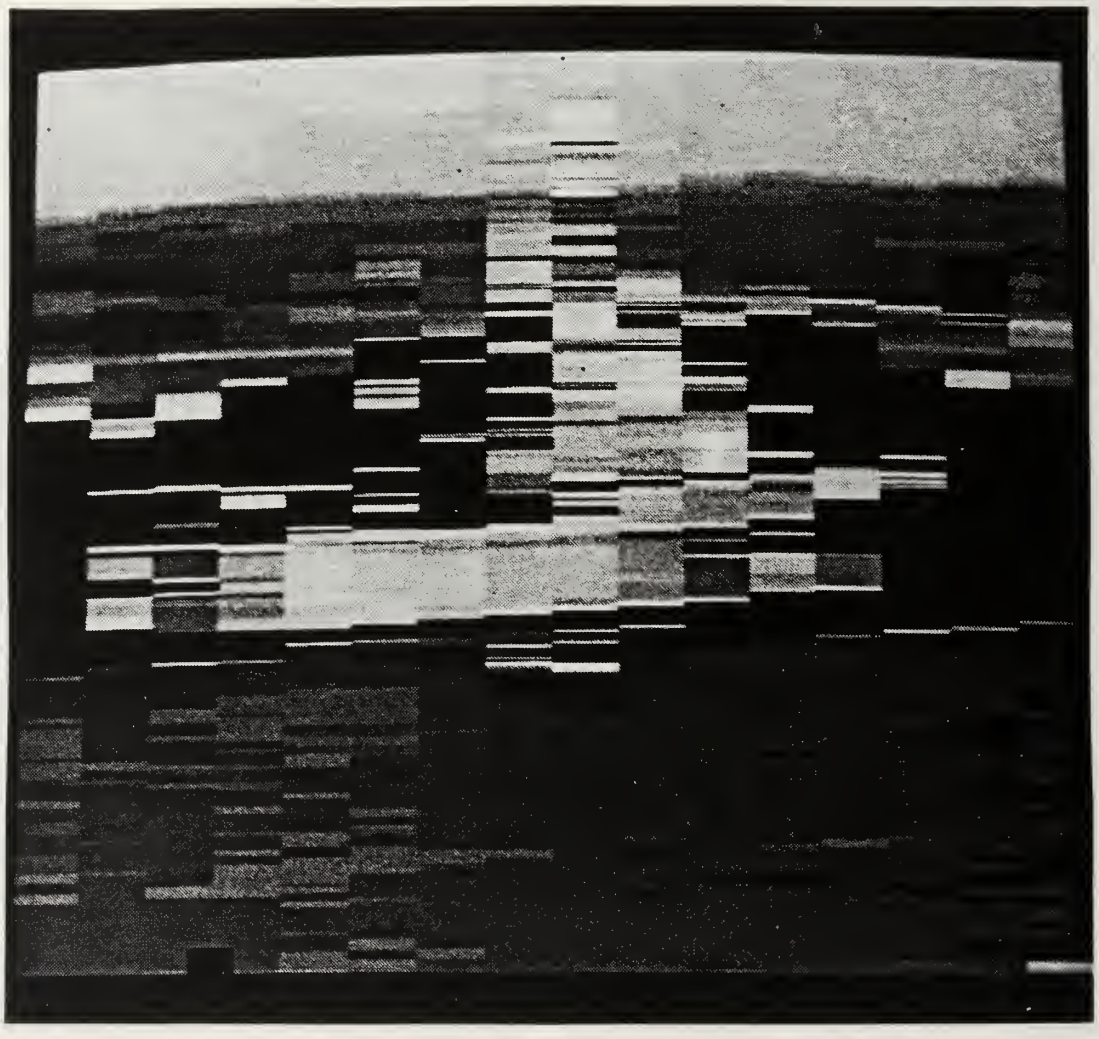

Figure D-6 Cutter16H

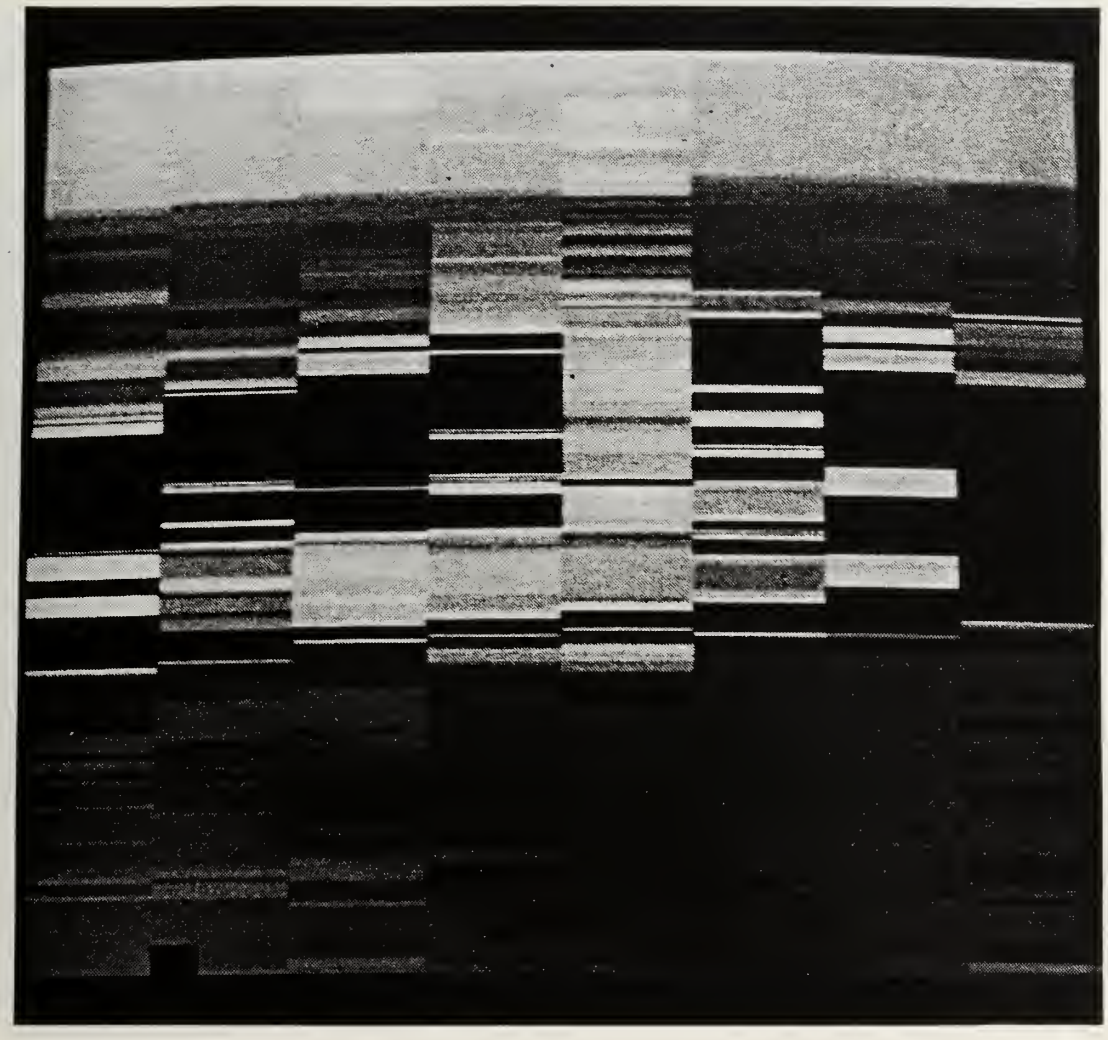

Figure D-7 Cutter8H

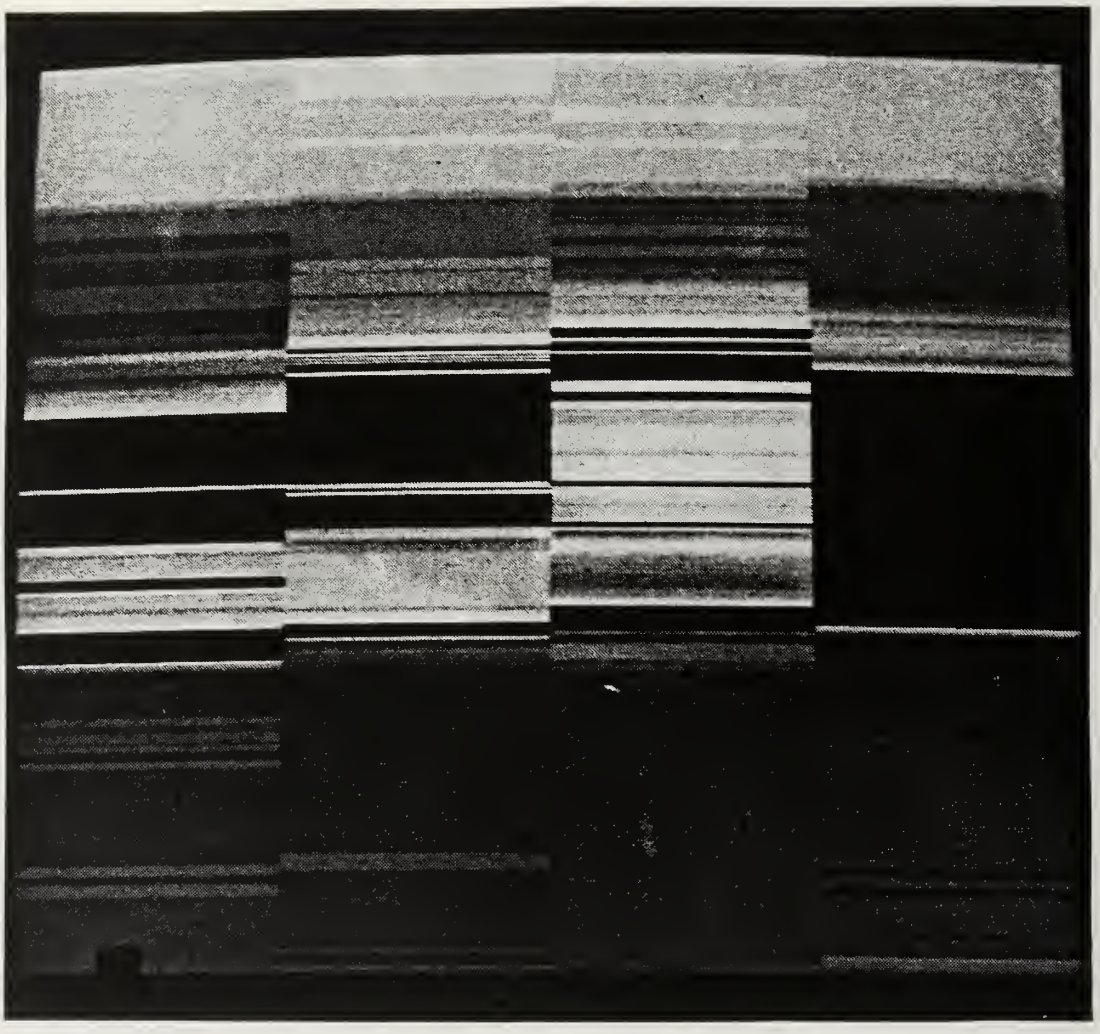

Figure D-8 Cutter4H

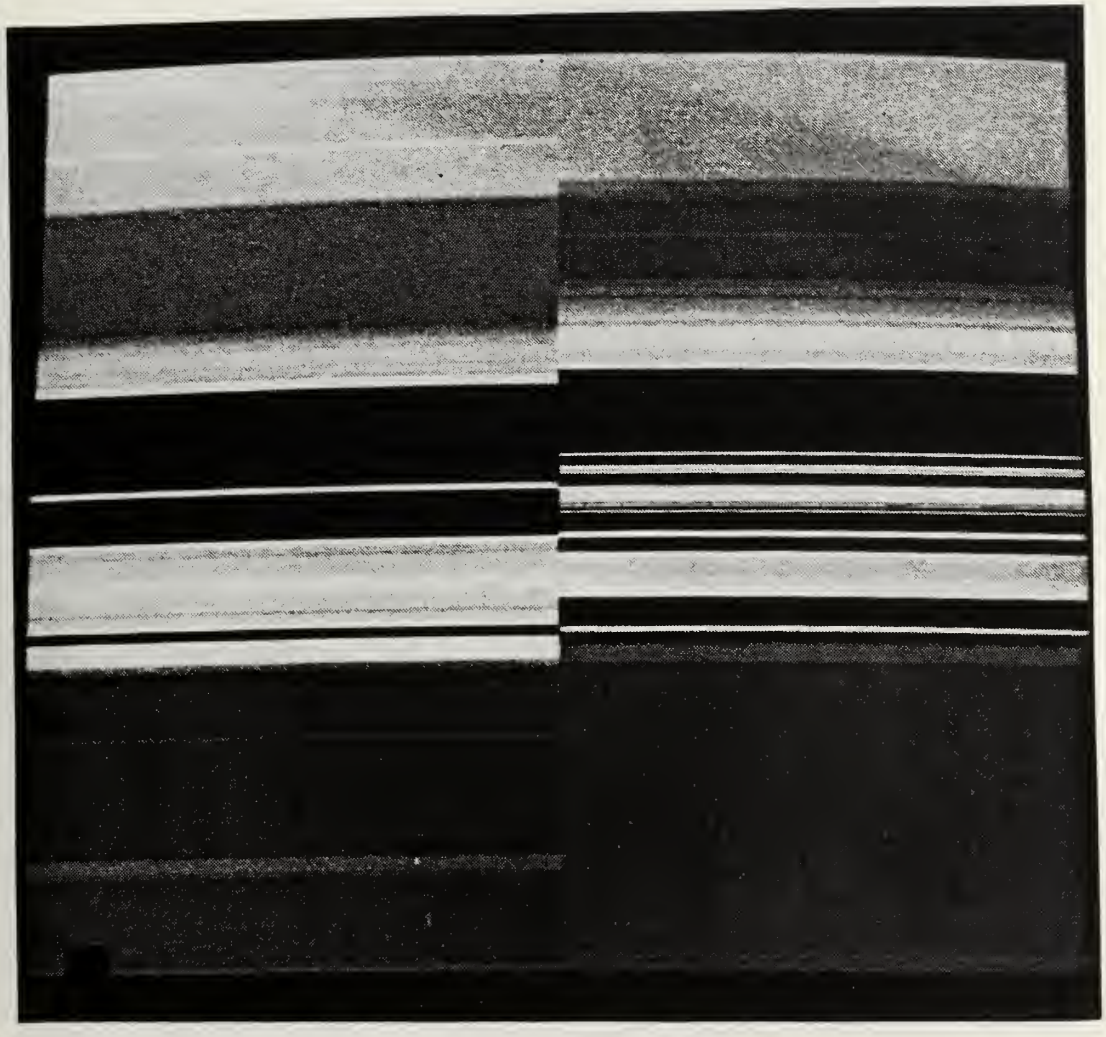

Figure D-9 Cutter2H

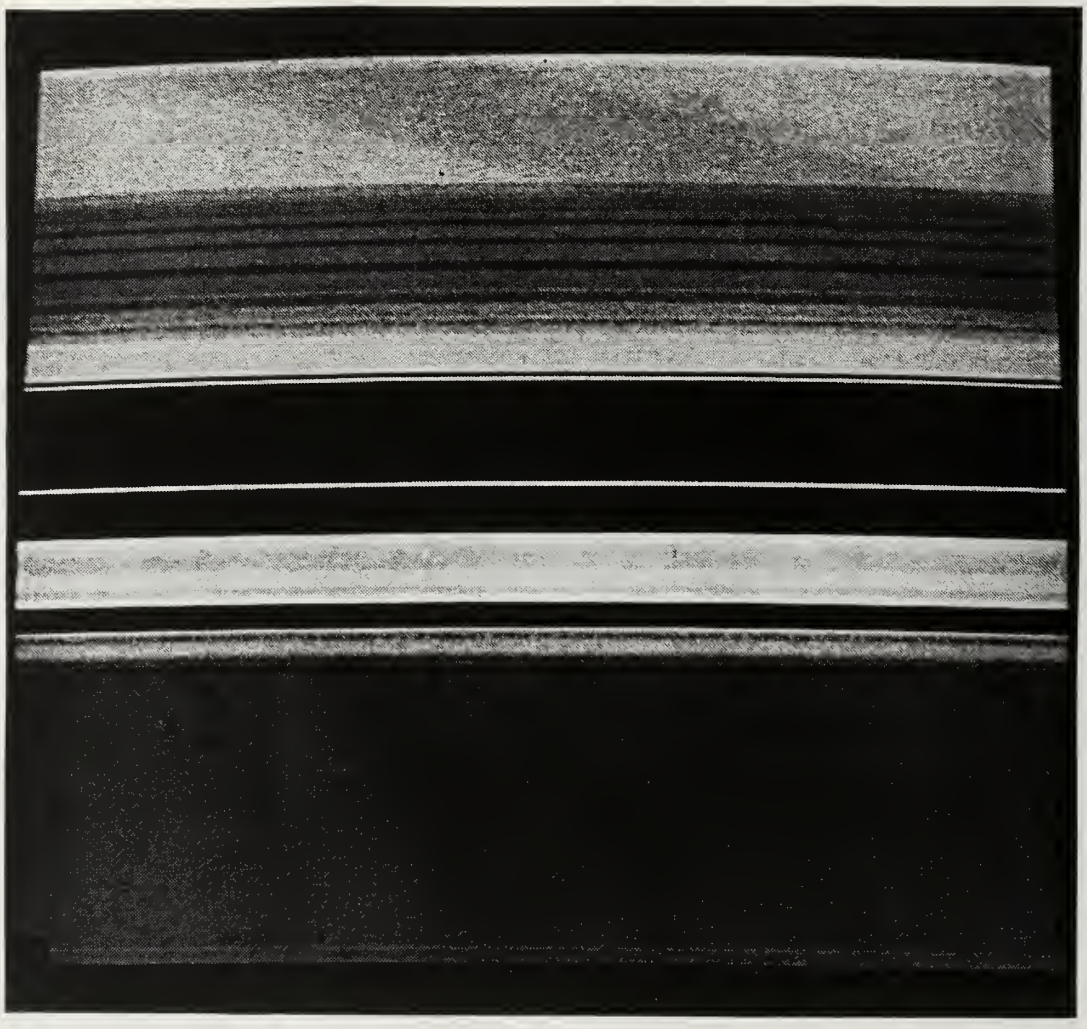

Figure D-10 Cutter1H

APPENDIX E

Figure E-1 Cutter512

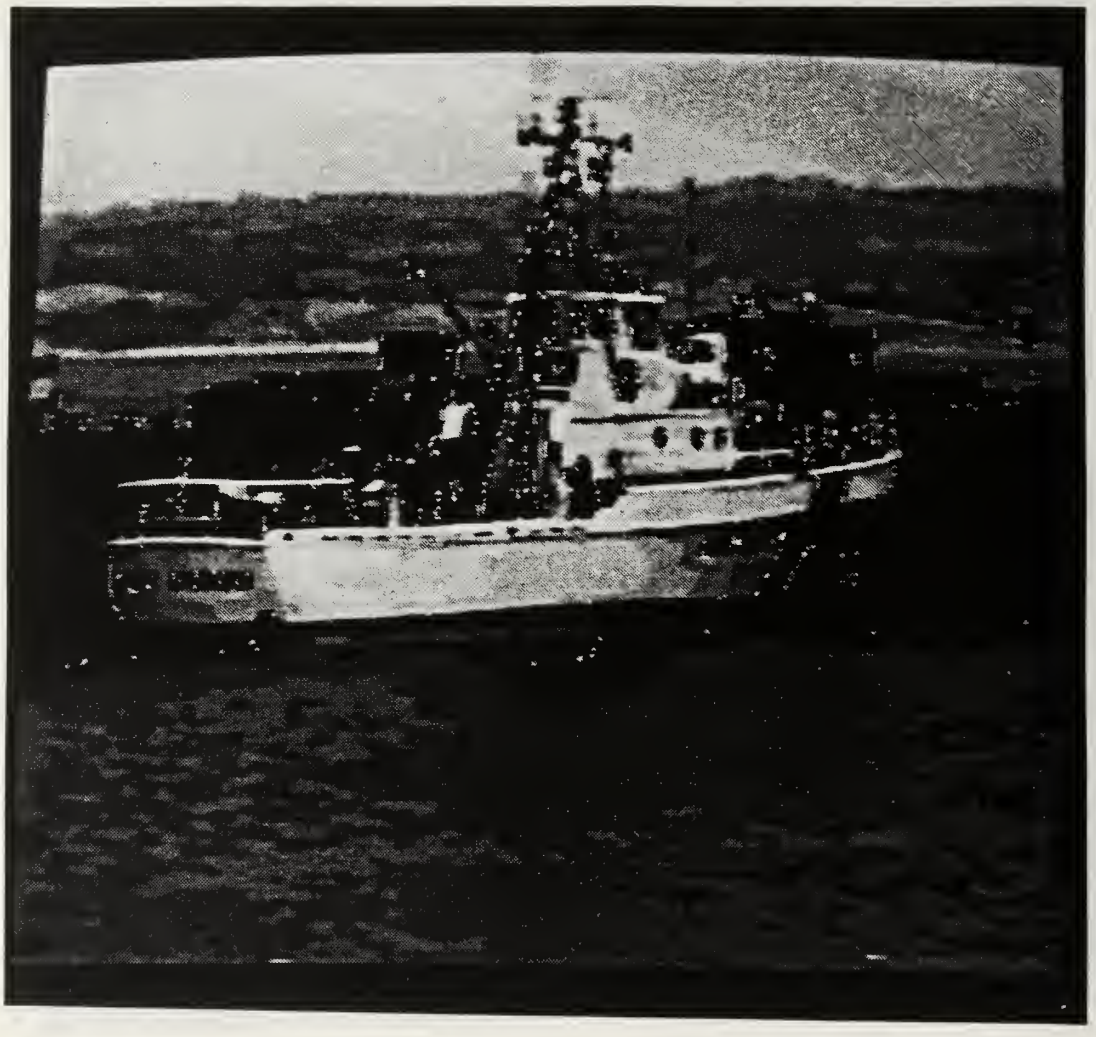

Figure E-2 Cutter256

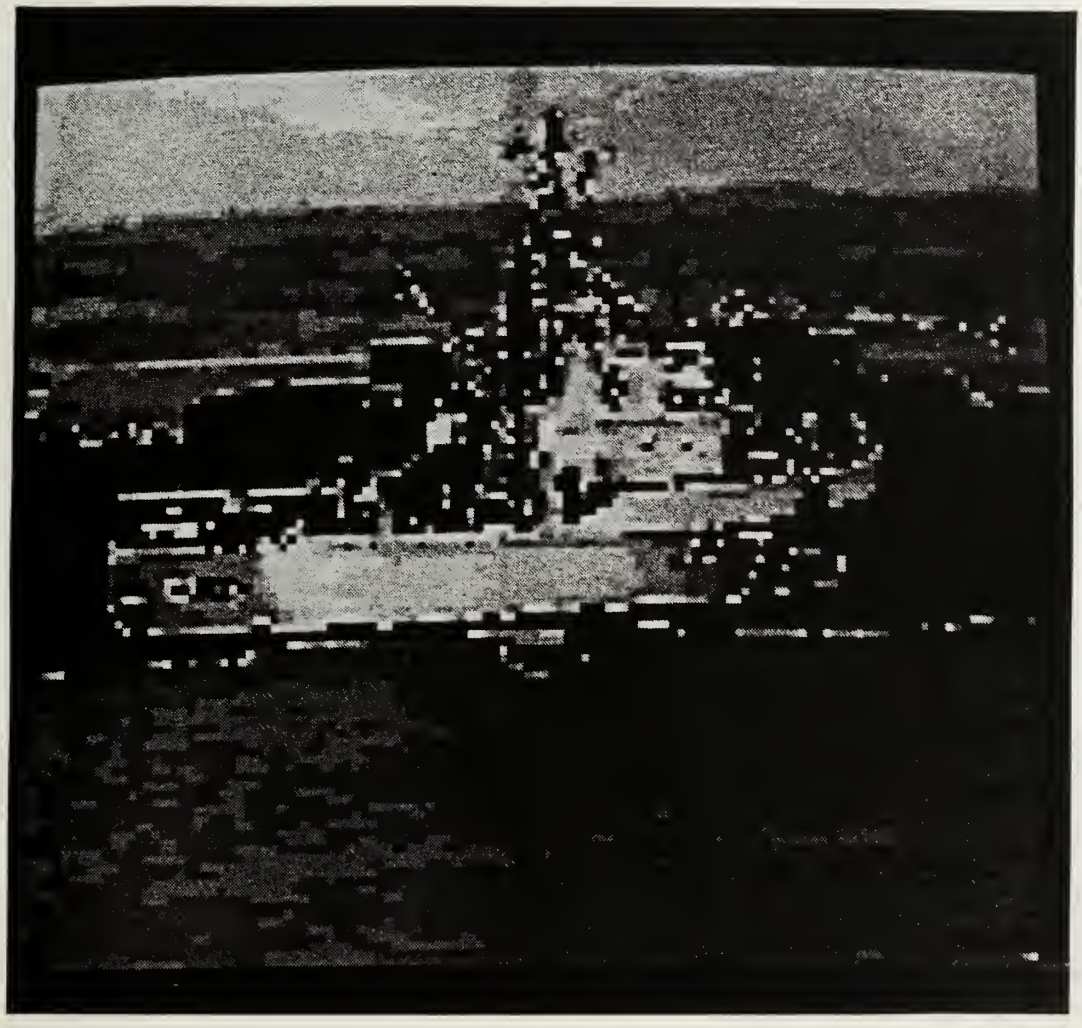

Figure E-3 Cutter128

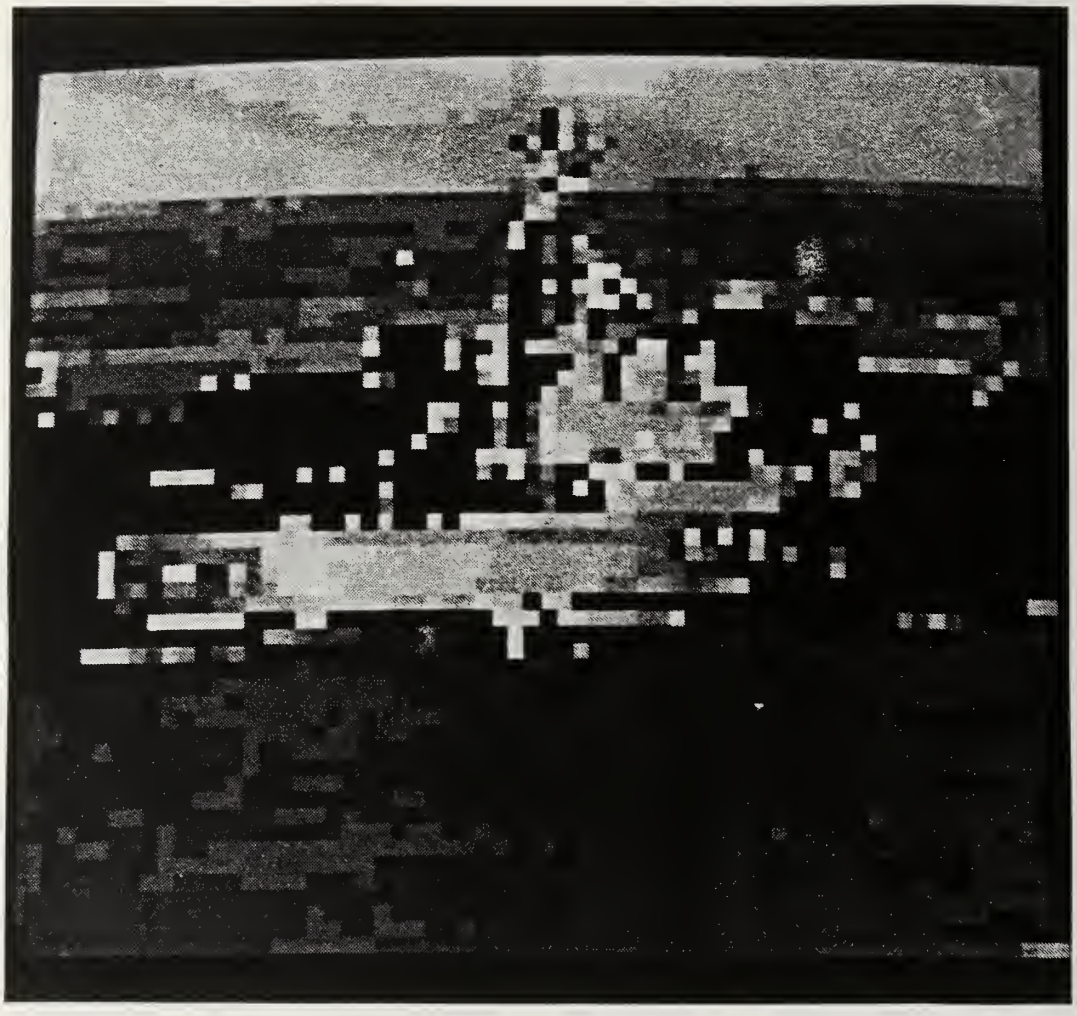

Figure E-4 Cutter64

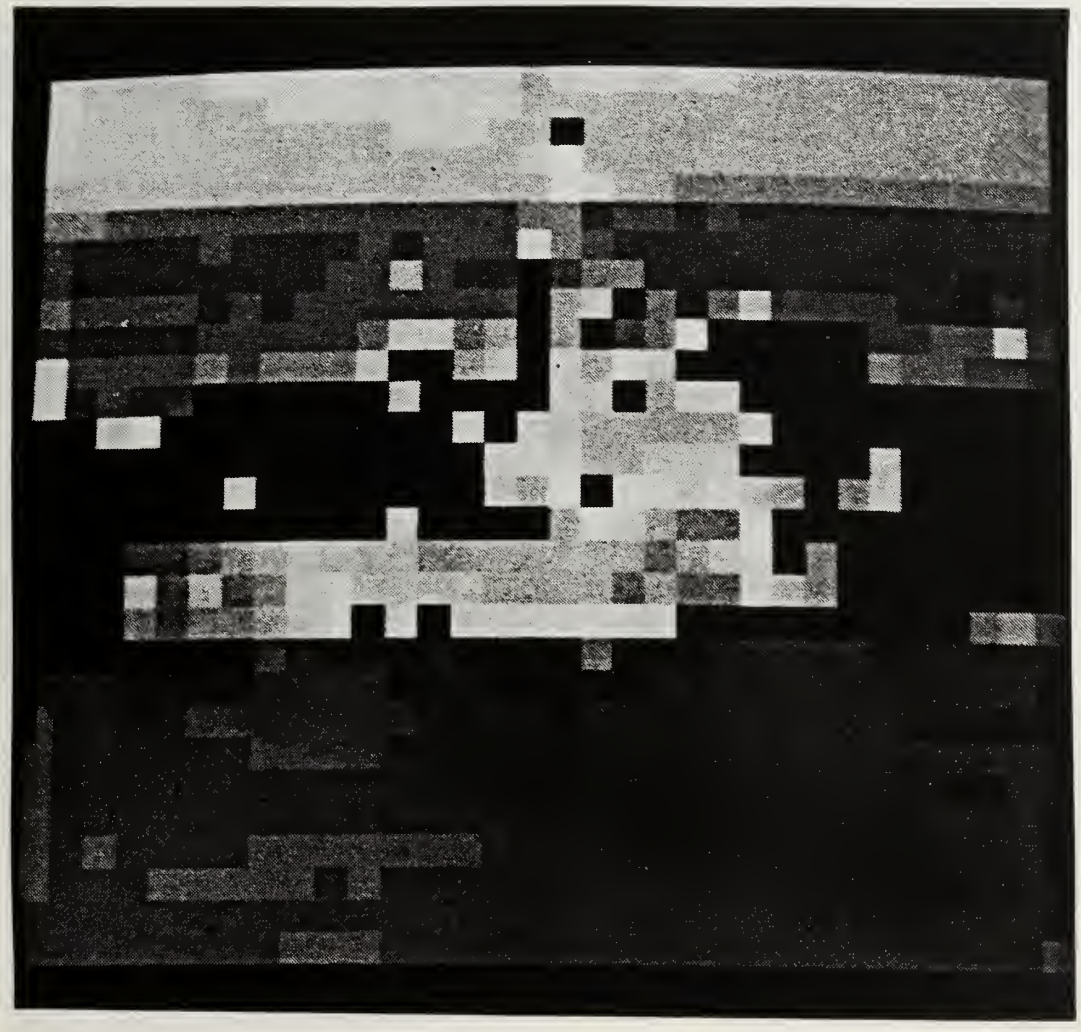

Figure E-5 Cutter32

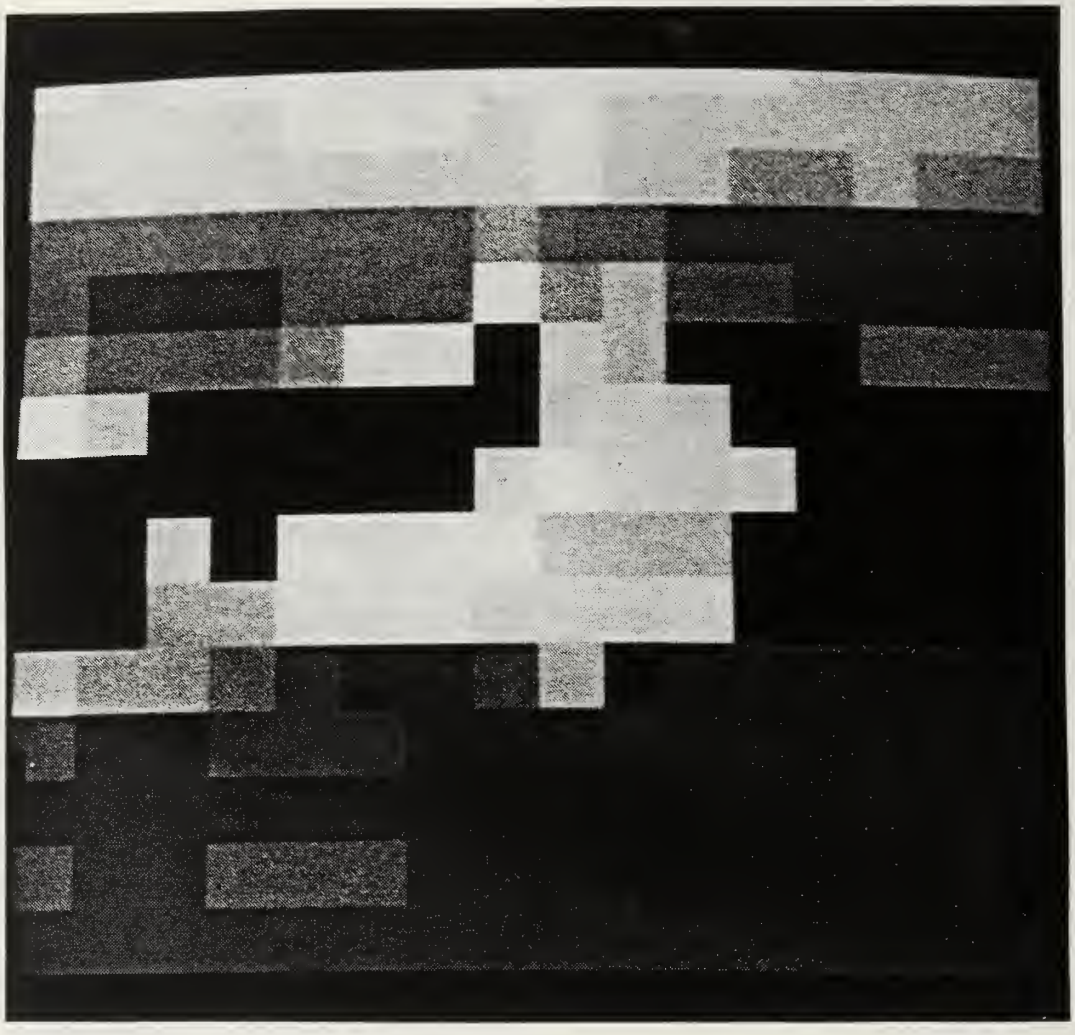

Figure E-6 Cutter16

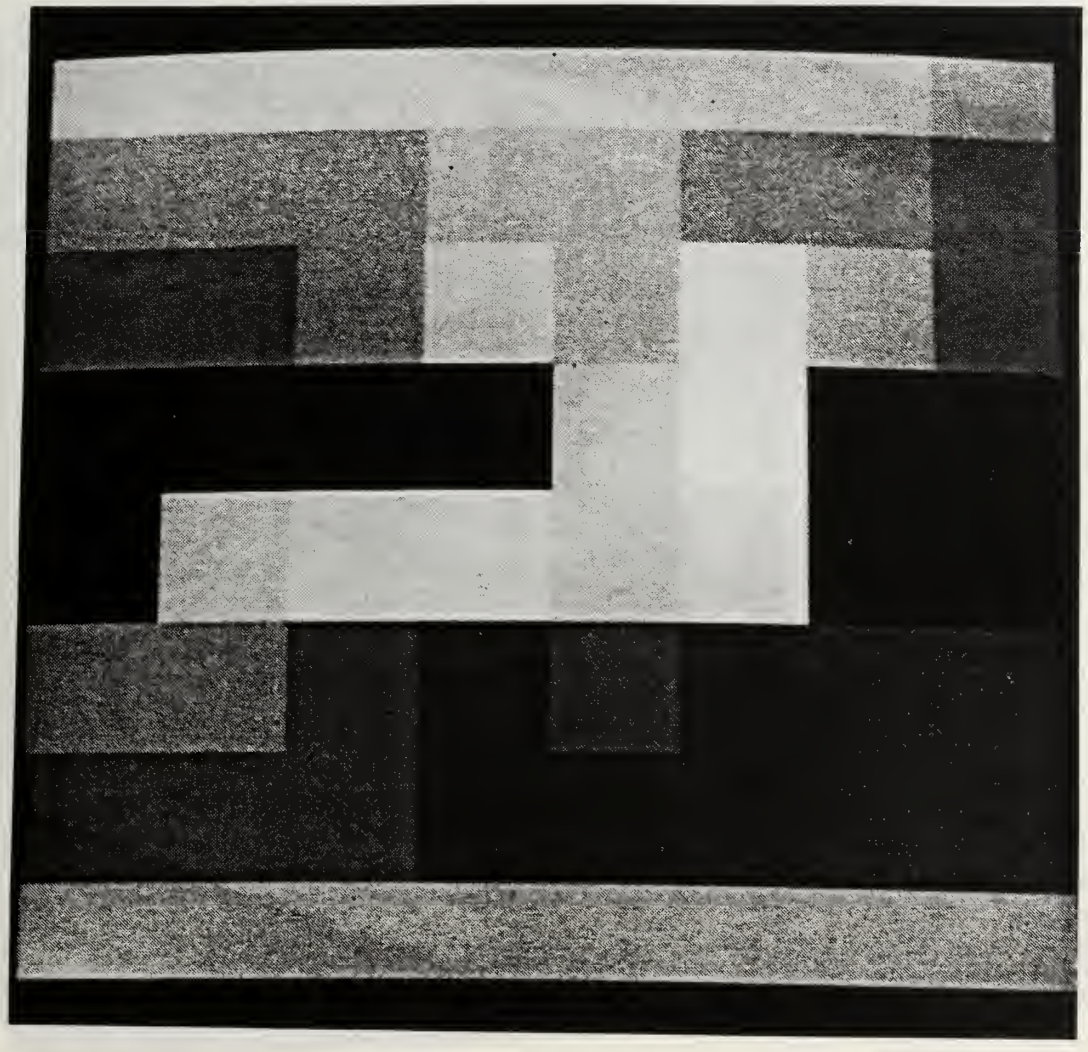

Figure E-7 Cutter8

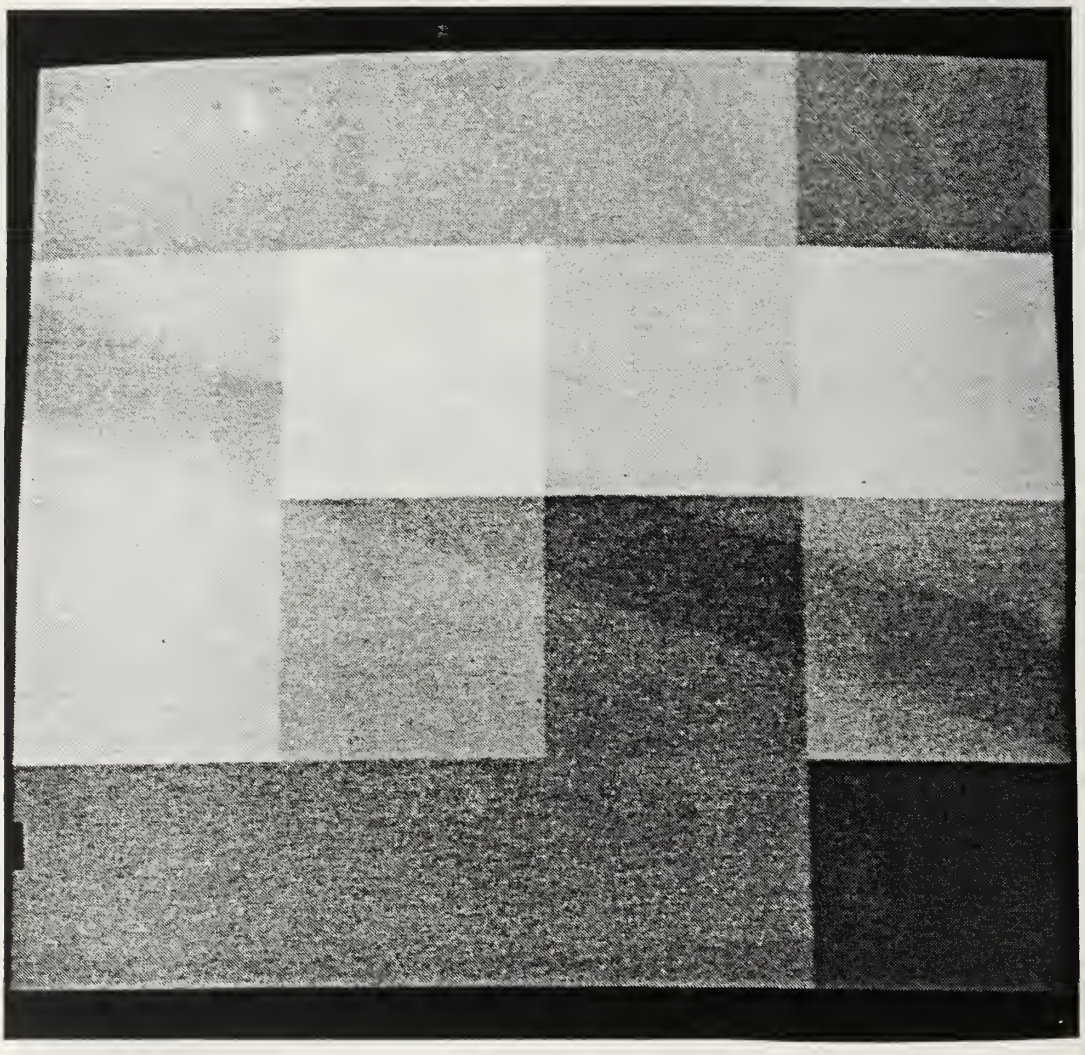

Figure E-8 Cutter4

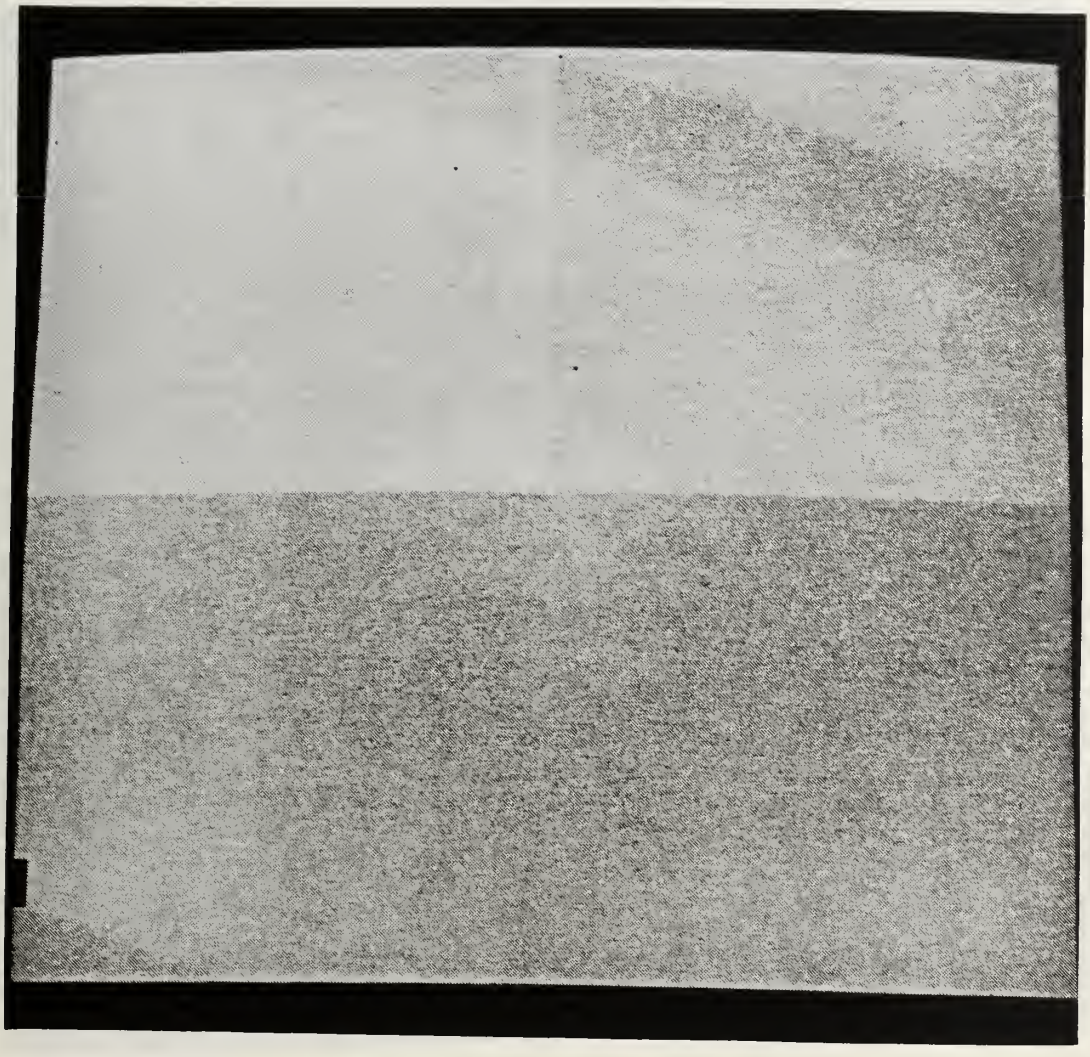

Figure E-9 Cutter2

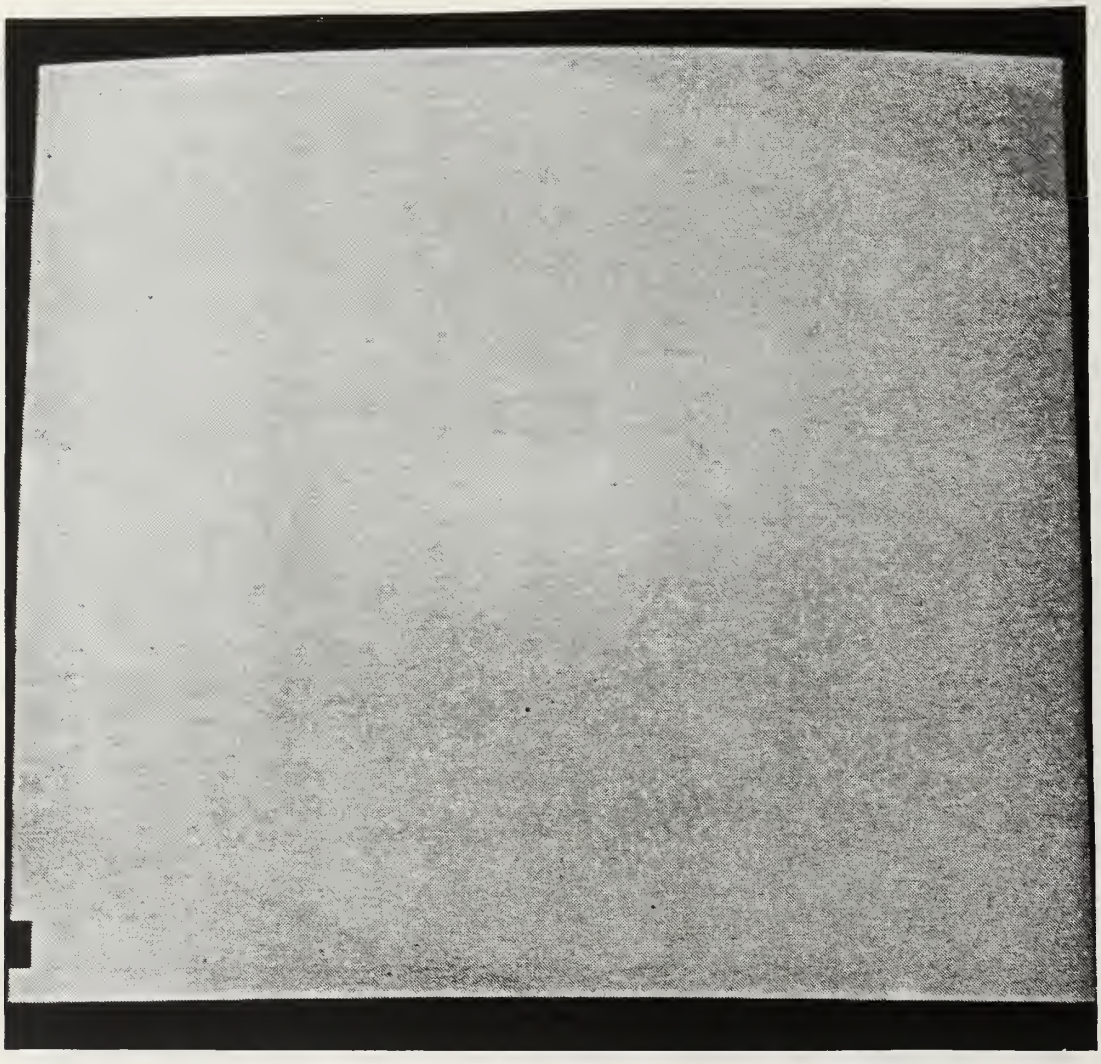

Figure E-10 Cutter1

APPENDIX F

Figure F-1 Lhouse512

Figure F-2 Lhouse256

Figure F-3 Lhouse128

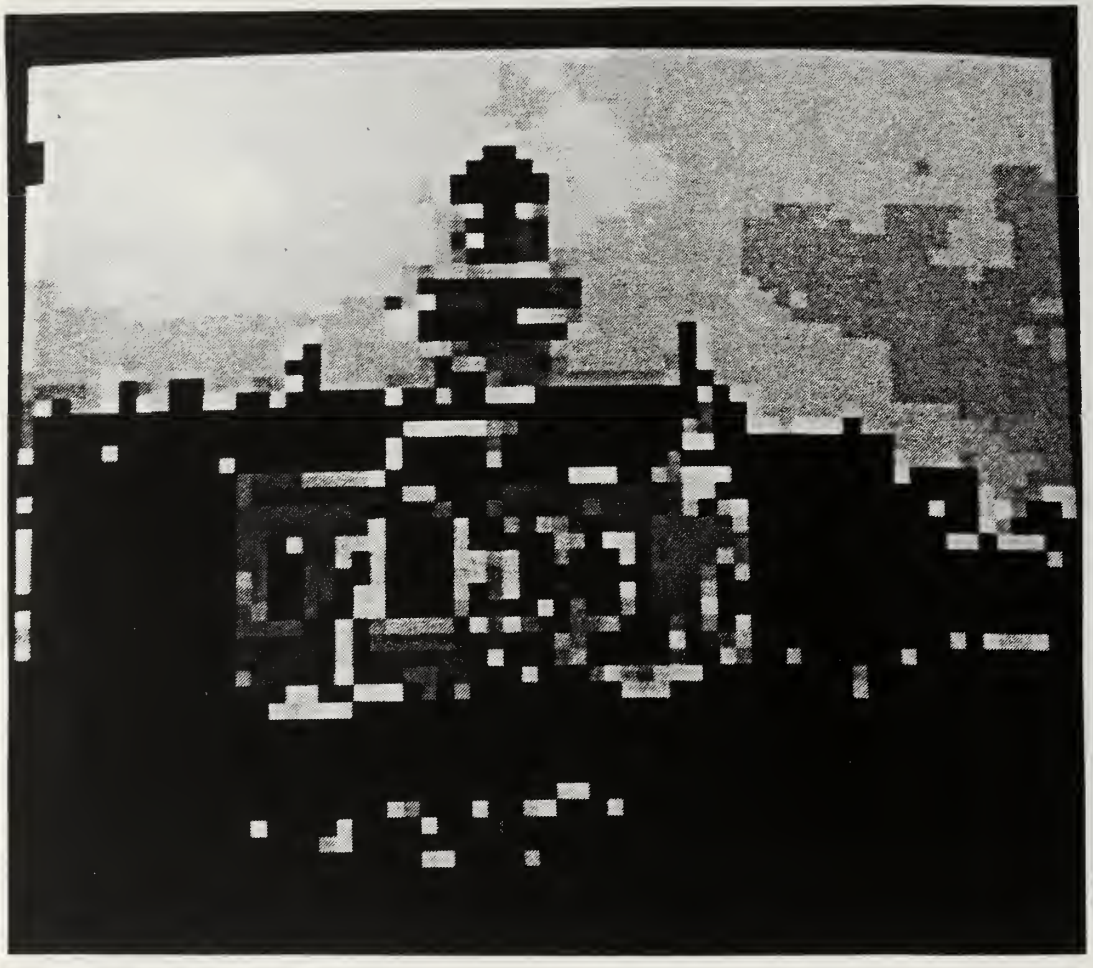

Figure F-4 Lhouse64

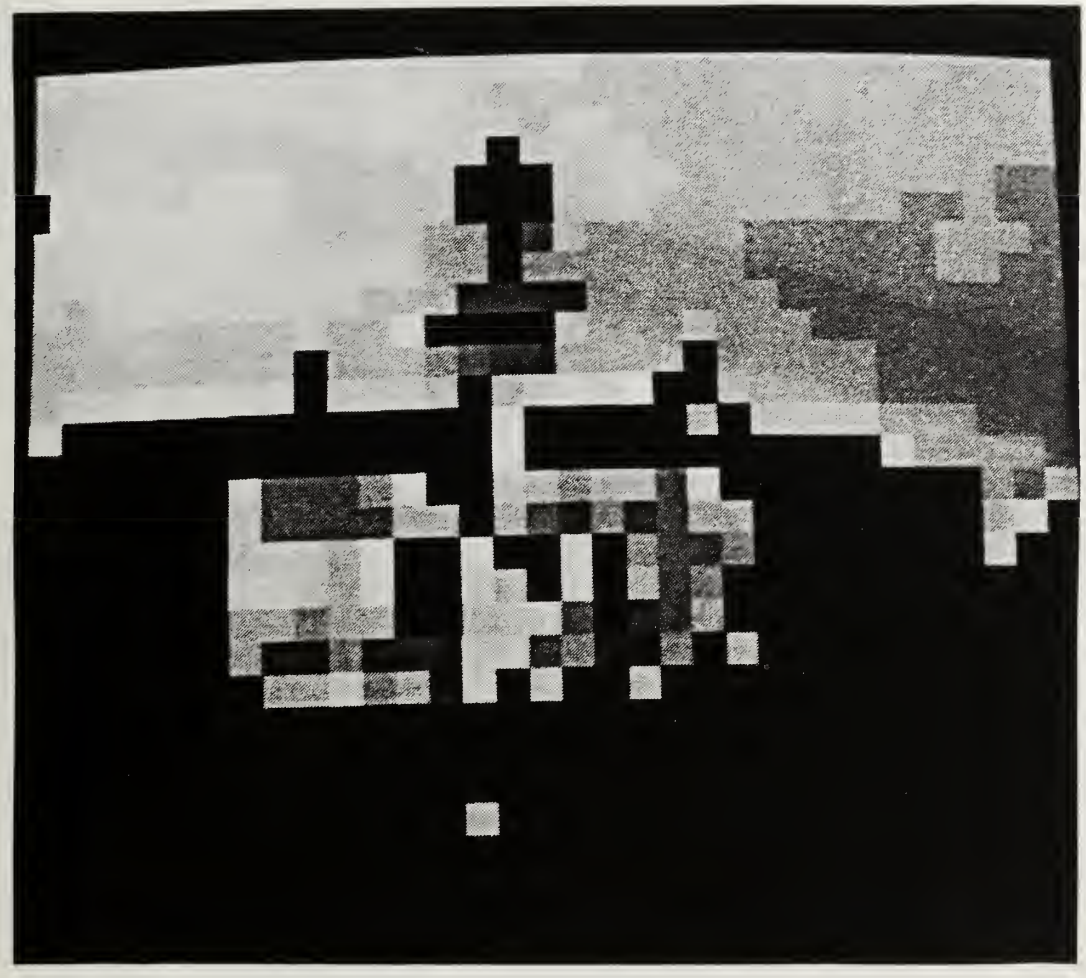

Figure F-5 Lhouse32

APPENDIX G

Figure G-1 Dog512

Figure G-2 Dog256

Figure G-3 Dog128

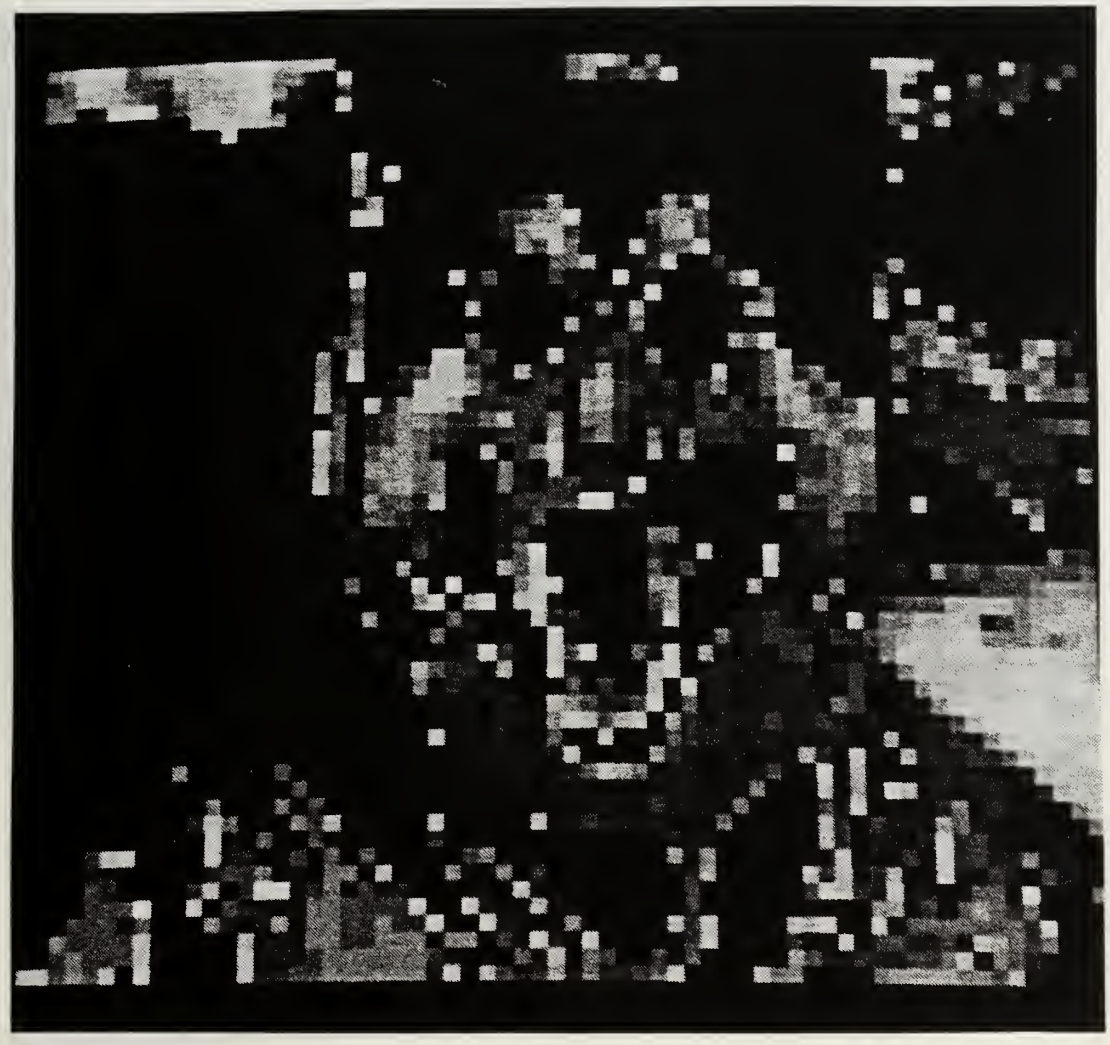

Figure G-4 Dog64

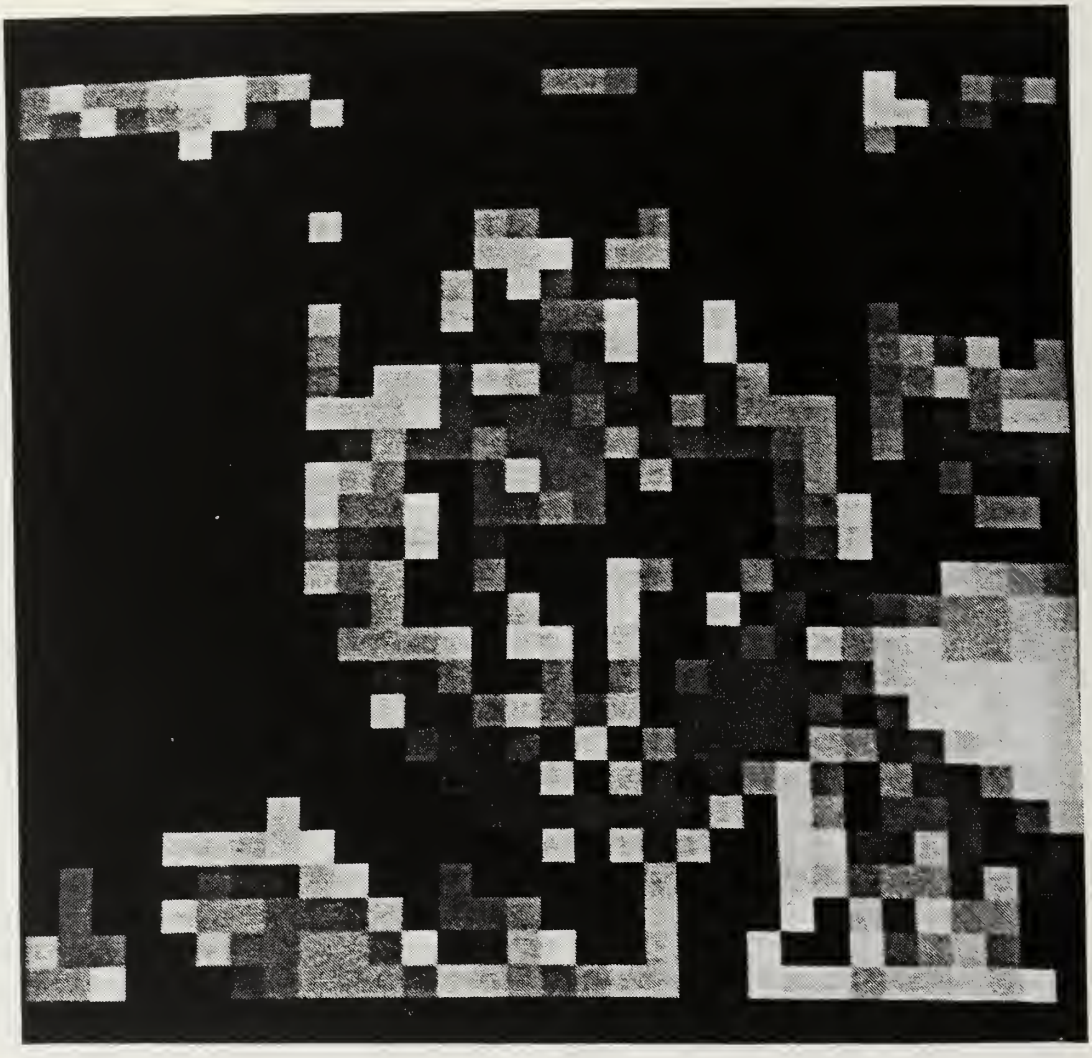

Figure G-5 Dog32

APPENDIX H

Figure H-1 Fence512

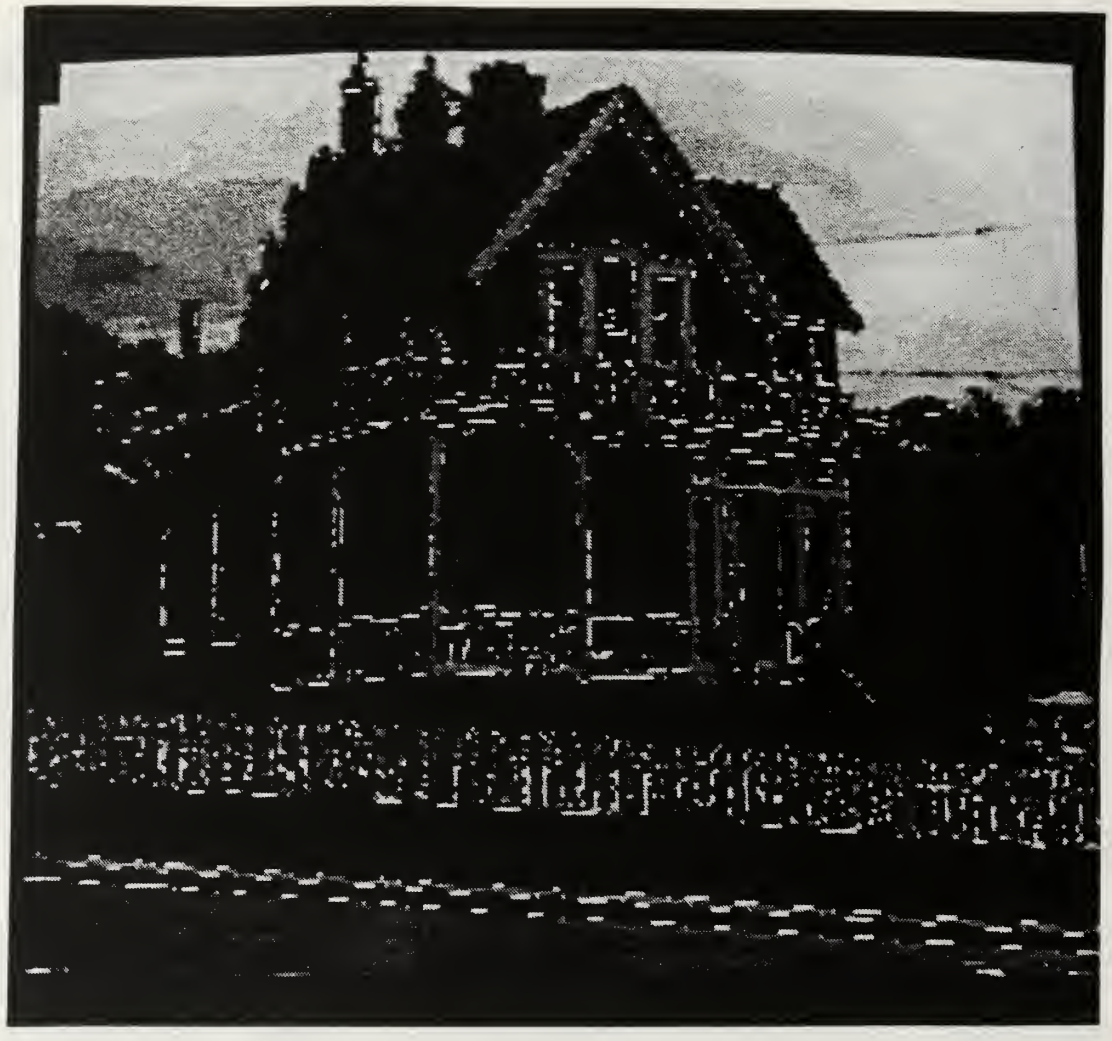

Figure H-2 Fence256

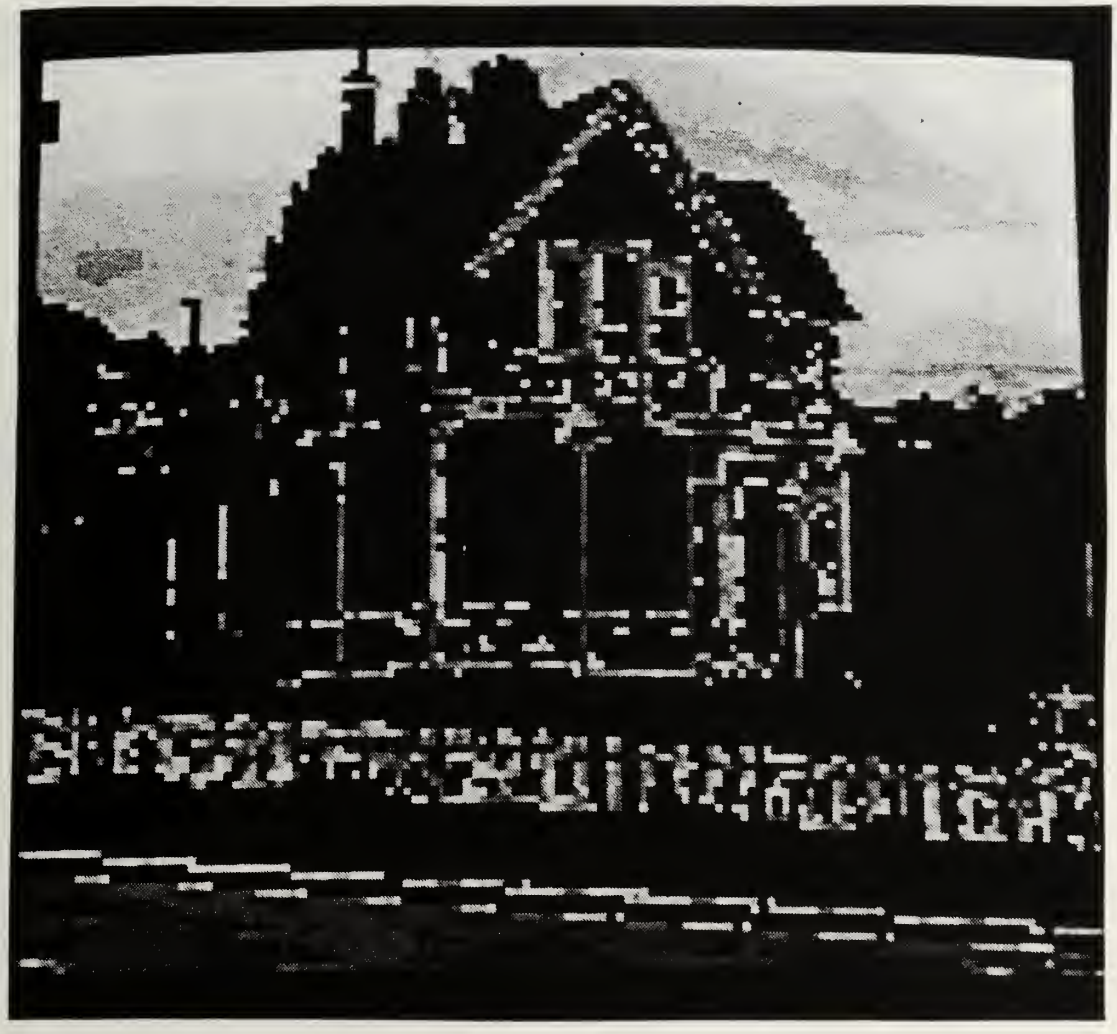

Figure H-3 Fence128

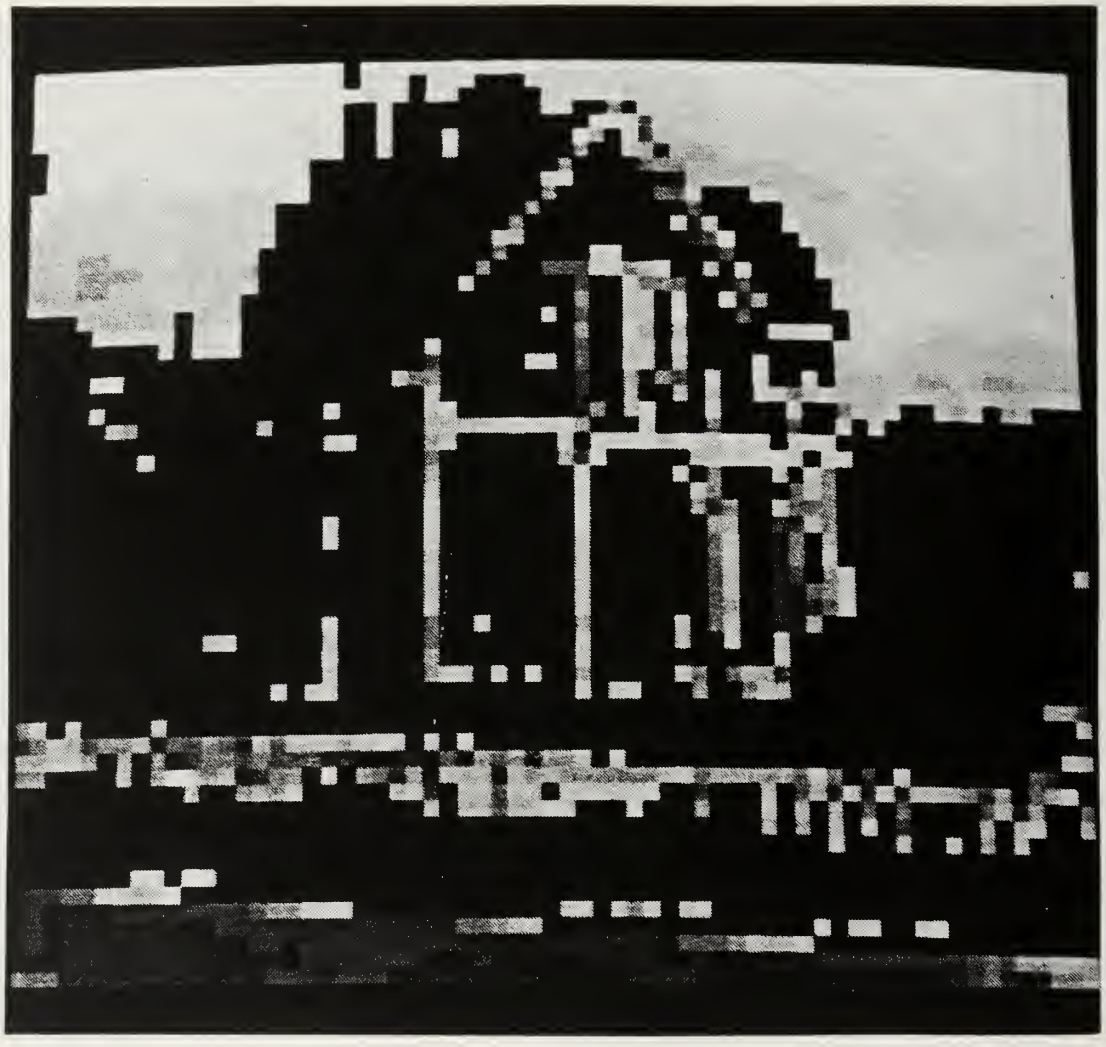

Figure H-4 Fence64

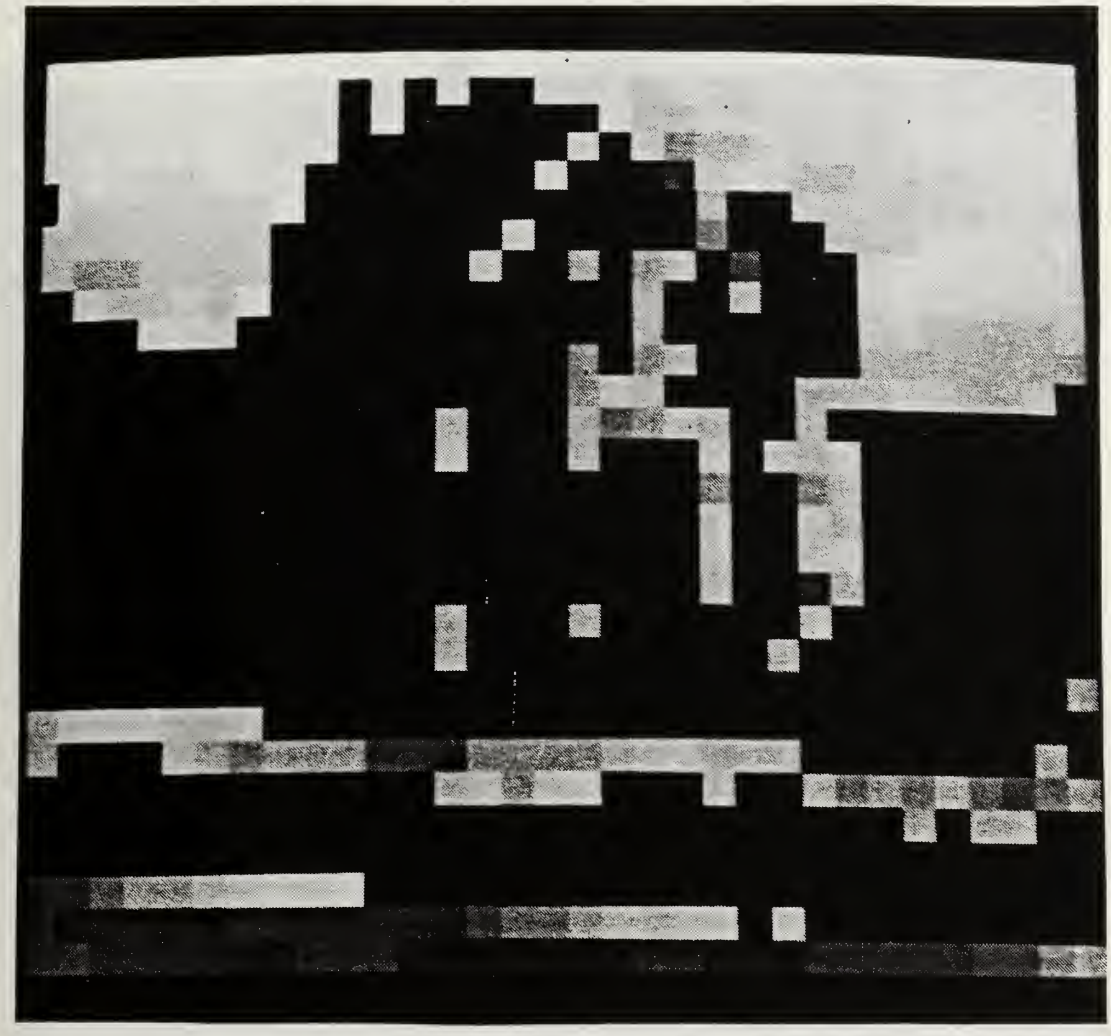

Figure H-5 Fence32

APPENDIX I

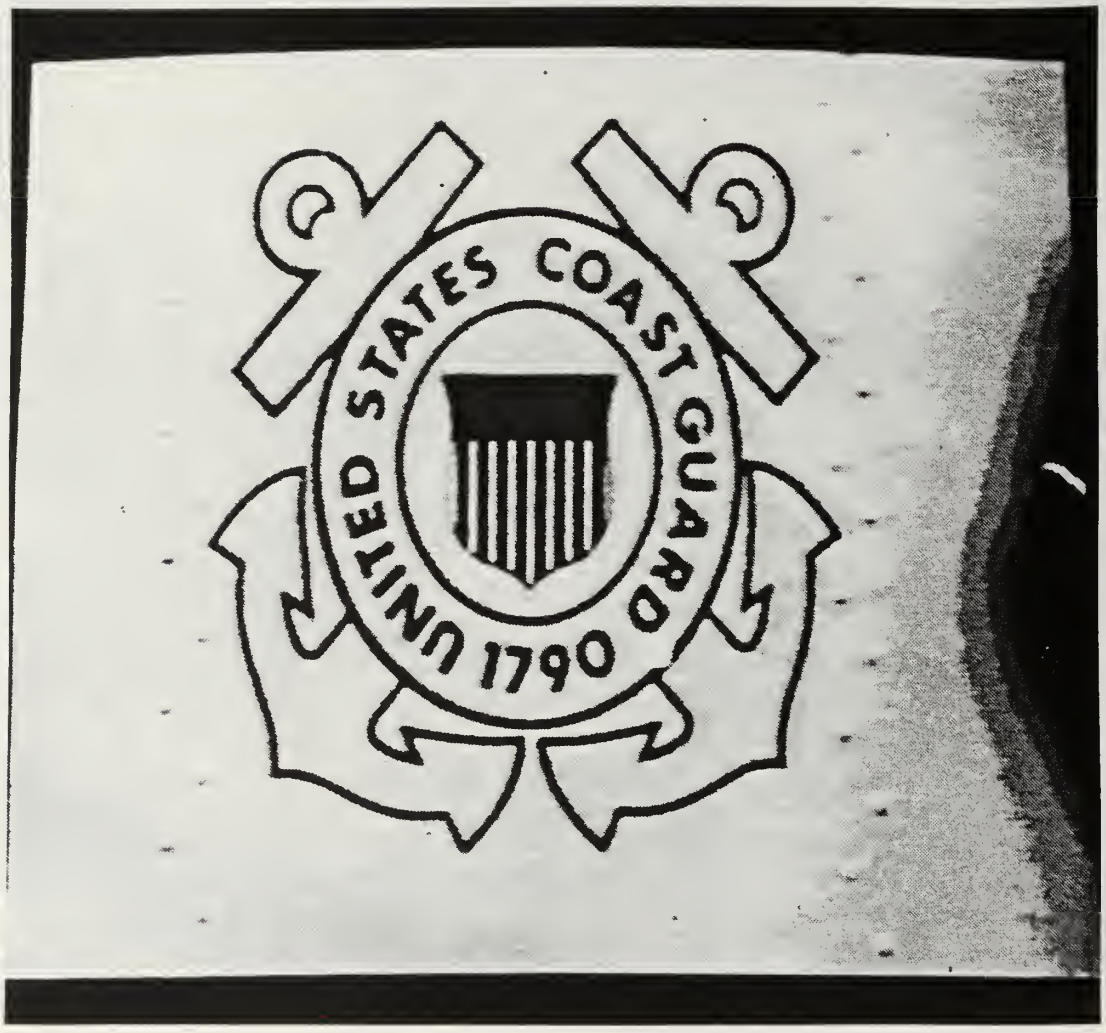

Figure I-1 Emblem512

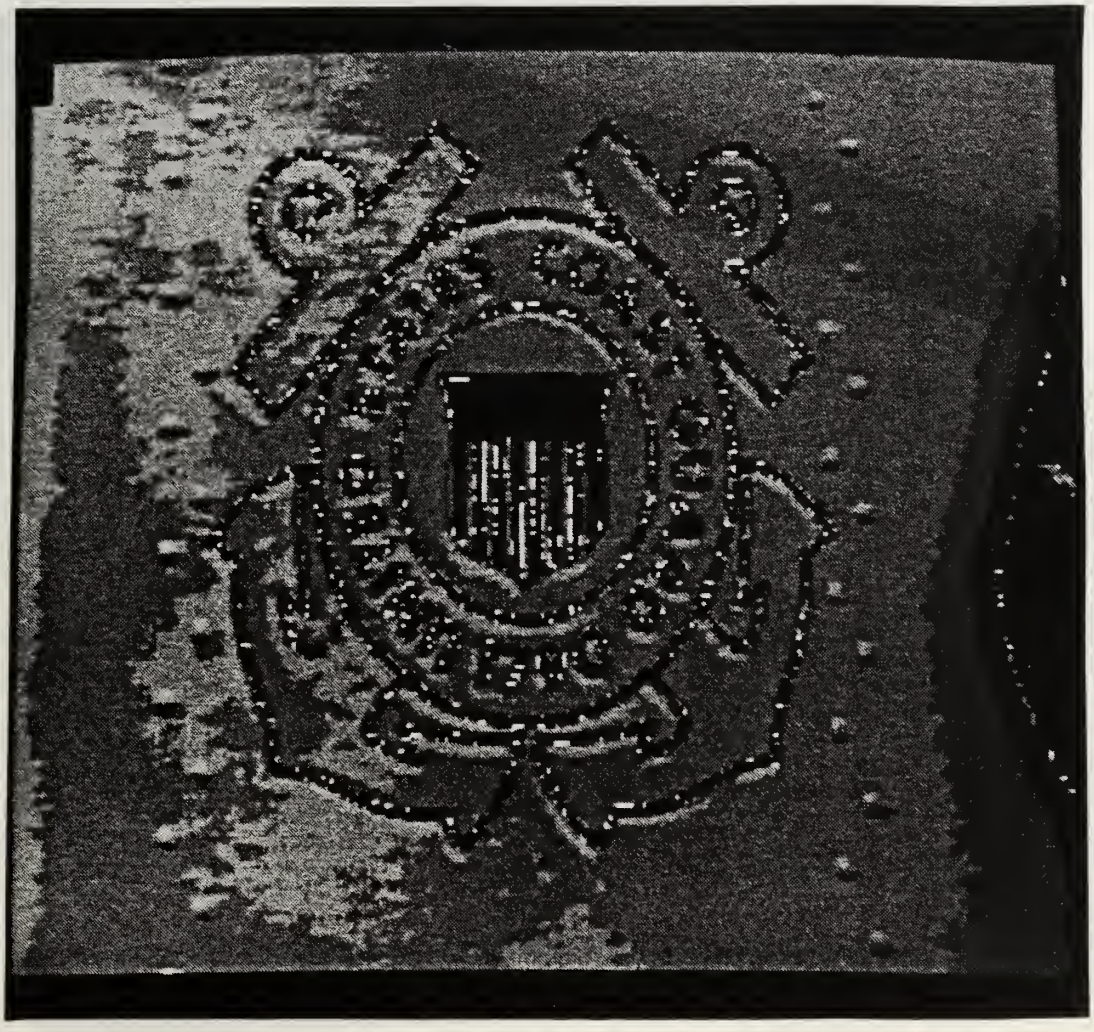

Figure I-2 Emblem256

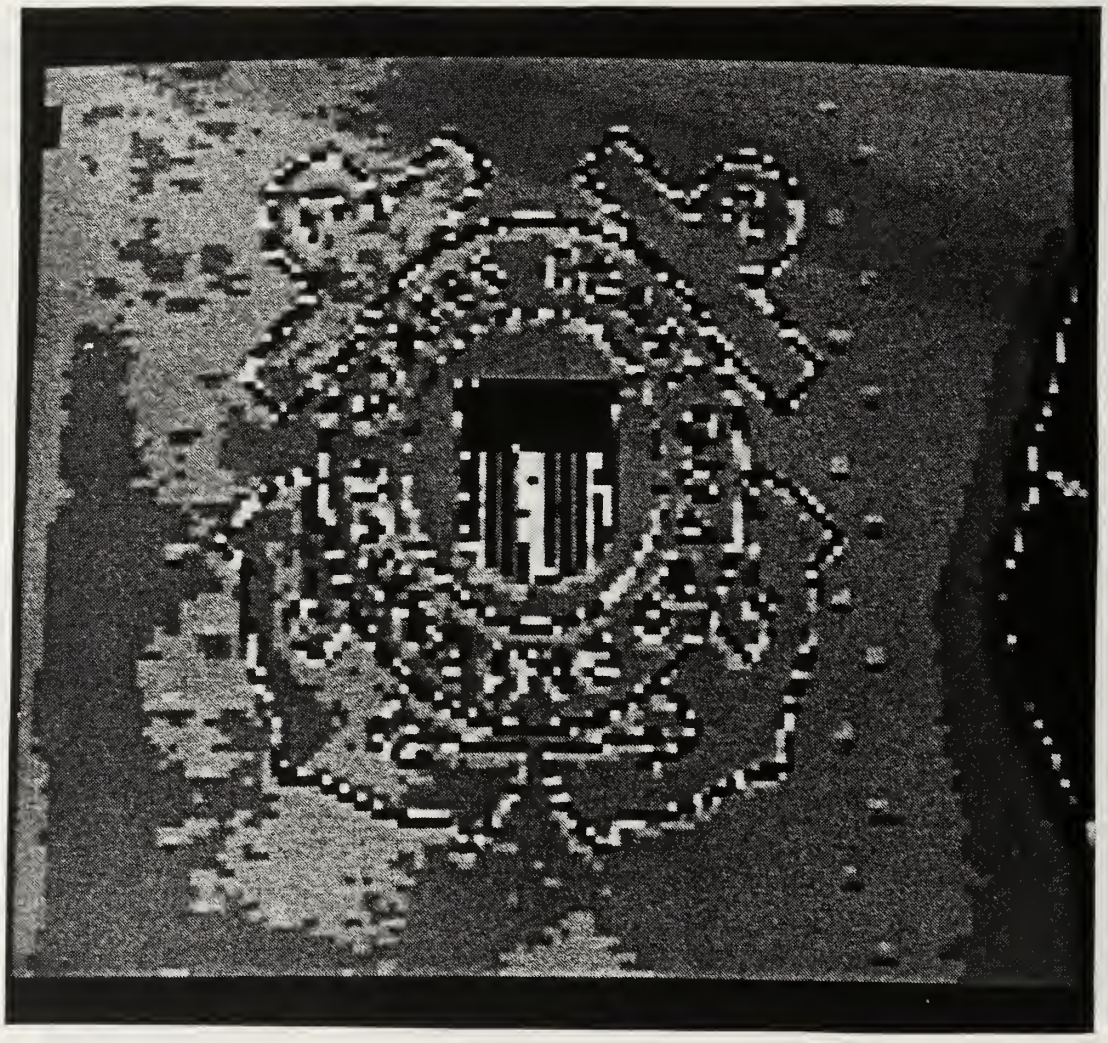

Figure I-3 Emblem128

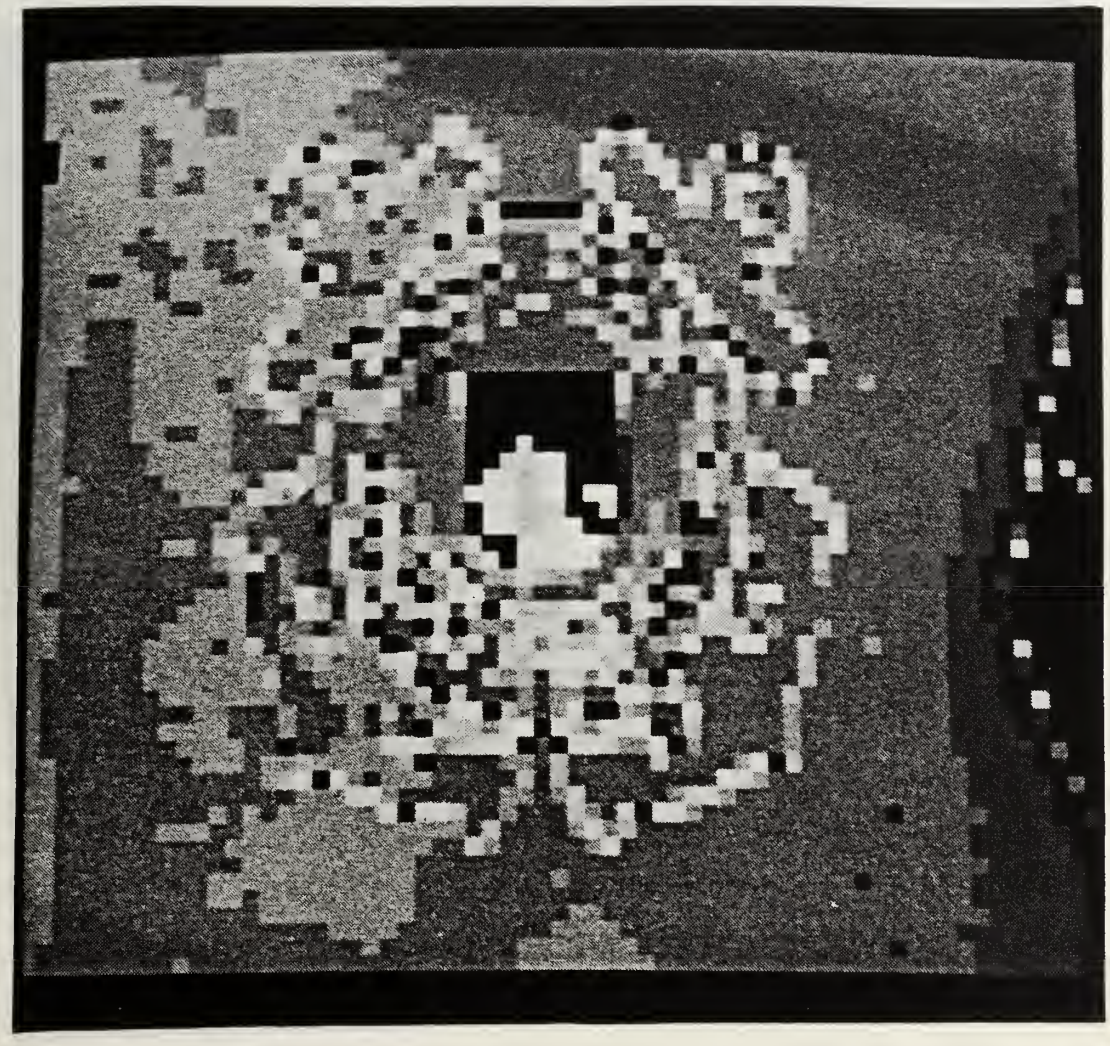

Figure I-4 Emblem64

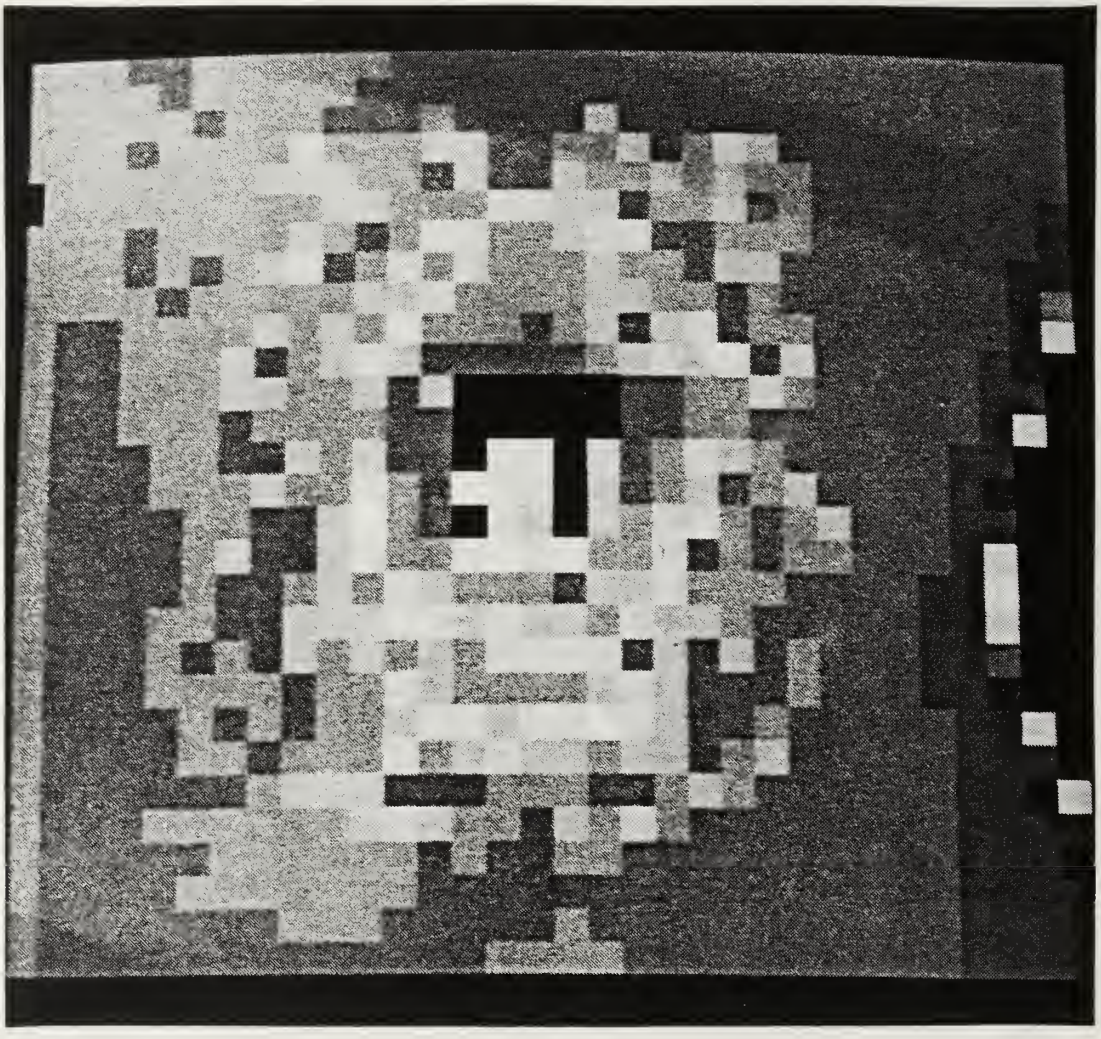

Figure I-5 Emblem32

APPENDIX J

```
/*****************************************************************************/
/*  C512: Conversion of 512 X 512 DT-IRIS Images into Simple Byte Format  */
/*  910614DSD    Microsoft C5.1                                              */
/*****************************************************************************/

#include <stdio.h>
#include <stdlib.h>

main (void)
  { unsigned int a, row, column;
    char instreamname[40], outstreamname[40];
    FILE * instream, * outstream;
    system ("CLS");
    for (a = 1; a <= 79; a ++)
      putch ('*');
    printf ("\n\r");
    for (a = 1; a <= 17; a ++)
      putch (' ');
    printf ("T512: 512 X 512 DT-IRIS Image Conversion\n\r");
    for (a = 1; a <= 79; a ++)
      putch ('*');
    printf ("\n\n\r");
    printf ("Name of DT-IRIS input file ? ");
    scanf ("%s", instreamname);
    if ((instream = fopen (instreamname, "rb")) == NULL)
      { fprintf (stderr, "Can't open input file!\n");
        exit (0); }
    printf ("\n\n\r");
    printf ("Name of translated output file ? ");
    scanf ("%s", outstreamname);
    if ((outstream = fopen (outstreamname, "wb")) == NULL)
      { fprintf (stderr, "Can't open output file!\n");
        exit (0); }
    printf ("\n\n\r");
    for (a = 1; a <= 512; a ++)
      fgetc (instream);
    for (row = 1; row <= 512; row ++)
      { for (column = 1; column <= 512; column ++)
        { a = fgetc (instream);
          fputc (a, outstream); }
        fputc ('\n', outstream); }
    fclose (instream);
    fclose (outstream);
    printf ("Output file contains 512 lines of 512 single-byte unsigned ");
    printf ("integers\n\n\n\n\r");
    return; }
```

APPENDIX K

```
/*************************************************************************
/*  D512: VGA Display for 512 X 512 Single-byte Greyscale Images
/*  910614DSD    Microsoft C5.1
/*************************************************************************

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <graph.h>
#include <math.h>

main (void)
  { int basis, a, amin, amax, row, column, rowoffset, xstart, y, x, color;
    long int tints[16];
    char instreamname[40];
    FILE * instream;
    struct videoconfig vc;
    basis = 512;
    tints[0] = 0x00000000;
    tints[1] = 0x00040404;
    tints[2] = 0x00080808;
    tints[3] = 0x000c0c0c;
    tints[4] = 0x00101010;
    tints[5] = 0x00141414;
    tints[6] = 0x00181818;
    tints[7] = 0x001c1c1c;
    tints[8] = 0x00202020;
    tints[9] = 0x00242424;
    tints[10] = 0x00282828;
    tints[11] = 0x002c2c2c;
    tints[12] = 0x00303030;
    tints[13] = 0x00343434;
    tints[14] = 0x00383838;
    tints[15] = 0x003c3c3c;
    system ("CLS");
    for (a = 1; a <= 79; a ++)
      putch ('*');
    printf ("\n\r");
    for (a = 1; a <= 20; a ++)
      putch (' ');
    printf ("D512: 512 X 512 Greyscale Image Display\n\r");
    for (a = 1; a <= 79; a ++)
      putch ('*');
    printf ("\n\n\r");
    printf ("Name of input file ? ");
    scanf ("%s", instreamname);
    if ((instream = fopen (instreamname, "rb")) == NULL)
      { fprintf (stderr, "Can't open input data file!\n");
        exit (0); }
    printf ("\n\n\r");
    amin = amax = 0;
    for (row = 1; row <= basis; row ++)
      { for (column = 1; column <= basis; column ++)
          { a = fgetc (instream);
            if (a < amin)
              amin = a;
            else if (a > amax)
              amax = a; }
        fgetc (instream); }
    rewind (instream);
    _setvideomode (_VRES16COLOR);
    _remapallpalette (tints);
    _getvideoconfig (& vc);
    _clearscreen (_GCLEARSCREEN);
    rowoffset = (basis - vc.numypixels) / 2;
    for (row = 1; row <= rowoffset; row ++)
```

```
  for (column = 1; column <= basis + 1; column ++)
    a = fgetc (instream);
xstart = (vc.numxpixels - basis) / 2;
y = 0;
for (row = 1; row <= vc.numypixels; row ++)
  { x = xstart;
    for (column = 1; column <= basis; column ++)
      { a = fgetc (instream);
        color = floor ((vc.numcolors - 1) * (a - amin) / (amax - amin) +
                + 0.5);
        _setcolor (color);
        _setpixel (x, y);
        ++ x; }
    fgetc (instream);
    ++ y; }
fclose (instream);
while (! kbhit ())
  ;
_clearscreen (_GCLEARSCREEN);
_setvideomode (_DEFAULTMODE);
return; }
```

APPENDIX L

```
*      LT B.J. MUSSELMAN
*
C      THESIS PROGRAM -  IMAGE.FOR

C      THIS PROGRAM IS CONFIGURED TO PERFORM A 2-DIMENSIONAL WALSH
C      TRANSFORM ON AN INPUT IMAGE FILE.  THE INVERSE TRANSFORM IS
C      THEN TAKEN AND THE TRANSFORMED DATA IS WRITTEN TO SEPARATE
C      OUTPUT IMAGE FILE.
*
       INTEGER*2 NUMROWS,NUMCOLS,MAXSEQ,CUTOFF,SEQUENCY,FILTER(513)
       INTEGER*2 ROW,COL
       INTEGER*1 BDAT(512,513)
       INTEGER*4 W(513)
*
       NUMROWS = 512
       NUMCOLS = 513
       MAXSEQ = NUMROWS - 1
*
       OPEN(11, FILE = 'B:\EMBLEM', FORM=' BINARY', STATUS = 'OLD')
       OPEN(12,FILE='C:\STUDENTS\BRIAN\IMAGE',FORM='BINARY',STATUS='OLD')
*
       WRITE(*,*) 'ELIMINATE SEQUENCY COEFFICIENTS BEYOND - '
       READ(*,*) CUTOFF
*
C      SEQUENCY FILTER DEFINED IN THIS SECTION
*
       DO 2 SEQ = 0, MAXSEQ
          FILTER(SEQ + 1) = 1
          IF(SEQ .GT. CUTOFF) FILTER(SEQ + 1) = 0
     2 CONTINUE
*
C      READ IMAGE FILE, ROW-BY-ROW INTO ARRAY BDAT
*
       DO 4 ROW = 1, NUMROWS
          DO 3 COL = 1, NUMCOLS
             READ(11) BDAT(ROW,COL)
     3    CONTINUE
     4 CONTINUE
*
C      THIS SECTION PERFORMS VERTICAL SEQUENCY FILTERING
*
       DO 8 COL = 1, NUMCOLS - 1
          WRITE(*,*) COL
          DO 5 ROW = 1, NUMROWS
             W(ROW) = BDAT(ROW,COL)
     5    CONTINUE
*
       CALL FWT (NUMROWS, W, .FALSE.)
*
          DO 6 SEQ = 0, MAXSEQ
             W(SEQ + 1) = W(SEQ + 1) * FILTER(SEQ + 1)
     6    CONTINUE
*
       CALL FWT (NUMROWS, W, .TRUE.)
*
          DO 7 ROW = 1, NUMROWS
             BDAT(ROW,COL) = W(ROW)
     7    CONTINUE
     8 CONTINUE
*
```

```
C     THIS SECTION PERFORMS HORIZONTAL SEQUENCY FILTERING
*
      DO 18 ROW = 1, NUMROWS
         WRITE(*,*) ROW
         DO 15 COL = 1, NUMCOLS
            W(COL) = BDAT(ROW,COL)
   15    CONTINUE
*
      CALL FWT ((NUMCOLS - 1), W, .FALSE.)
*
         DO 16 SEQ = 0, MAXSEQ
            W(SEQ + 1) = W(SEQ + 1) * FILTER(SEQ + 1)
   16    CONTINUE
*
      CALL FWT ((NUMCOLS - 1), W, .TRUE.)
*
         DO 17 COL = 1, NUMCOLS - 1
            BDAT(ROW,COL) = W(COL)
   17    CONTINUE
   18 CONTINUE
*
C     WRITE IMAGE FILE, ROW-BY-ROW
*
      DO 11 ROW = 1, NUMROWS
         DO 19 COL = 1, NUMCOLS
            WRITE(12) BDAT(ROW,COL)
   19    CONTINUE
   11 CONTINUE
*
      CLOSE (11)
      CLOSE (12)
      STOP
      END
*
      SUBROUTINE FWT (NUMSAMPLS,DATARRAY,ENABLESCALE)
C     FAST WALSH TRANSFORM (TRANSLATED FROM D.S.D.'S VERSION IN "C")
C
C     NUMSAMPLS = NUMBER OF SEQUENCIES AND SAMPLING INTERVALS
C                 IN WALSH BASIS (INTEGER)
C
C     DATARRAY = INPUT/OUTPUT ARRAY (REAL)
C
C     ENABLESCALE = TRANSFORM SCALING ENABLE SWITCH
C
      INTEGER*2 NUMSAMPLS
      INTEGER*4 DATARRAY(1)
      LOGICAL ENABLESCALE
*
      CALL BITREVSORT (NUMSAMPLS,DATARRAY)
*
      CALL DECINSEQ (NUMSAMPLS,DATARRAY)
*
      IF (ENABLESCALE) CALL SCALE (NUMSAMPLS,DATARRAY)
      RETURN
      END
*
      SUBROUTINE BITREVSORT (NUMSAMPLS,DATARRAY)
*
C     BINARY BIT - REVERSAL SORT
*
```

```
      INTEGER NUMSD2,DIRECT,REVERSED,DATAD,DATAR,OFFSET
      INTEGER*2 NUMSAMPLS
      REAL SCRATCH
      INTEGER*4 DATARRAY(1)
*
      NUMSD2 = NUMSAMPLS / 2
      REVERSED = 0
      DO 30 DIRECT = 0, NUMSAMPLS - 1
         IF (DIRECT .GE. REVERSED) GO TO 10
            DATAD = 1 + DIRECT
            DATAR = 1 + REVERSED
            SCRATCH = DATARRAY(DATAD)
            DATARRAY(DATAD) = DATARRAY(DATAR)
            DATARRAY(DATAR) = SCRATCH
   10    OFFSET = NUMSD2
   20    IF((OFFSET .GT. REVERSED) .OR. (OFFSET .LT. 2)) GO TO 30
            REVERSED = REVERSED - OFFSET
            OFFSET = OFFSET / 2
            GO TO 20
   30 REVERSED = REVERSED + OFFSET
      RETURN
      END
*
      SUBROUTINE DECINSEQ (NUMSAMPLS,DATARRAY)
*
C     DECIMATION IN SEQUENCE
*
      INTEGER LOG2NUMSAMPLS,MSB,BLOCKSIZE,STAGE
      INTEGER NODE1START,NODE2START,NODE1,NODE2
      INTEGER*2 NUMSAMPLS
      INTEGER*4 DATARRAY(1)
      LOGICAL TOGGLE
      REAL SCRATCH
*
      LOG2NUMSAMPLS = 0
      MSB = NUMSAMPLS
   10 MSB = MSB / 2
      IF (MSB .EQ. 0) GO TO 20
         LOG2NUMSAMPLS = LOG2NUMSAMPLS + 1
         GO TO 10
   20 BLOCKSIZE = NUMSAMPLS / 2
      DO 80 STAGE = 1, LOG2NUMSAMPLS
         TOGGLE = .FALSE.
         NODE1START = 0
   30    IF (NODE1START .GE. NUMSAMPLS) GO TO 80
            NODE2START = NODE1START + BLOCKSIZE
            NODE2 = NODE2START
            IF (.NOT. TOGGLE) GO TO 50
               DO 40 NODE1 = NODE1START,NODE2START - 1
                  SCRATCH = DATARRAY(NODE1 + 1)
                  DATARRAY(NODE1+1)=DATARRAY(NODE1+1)-DATARRAY(NODE2+1)
                  DATARRAY(NODE2+1) = DATARRAY(NODE2+1) + SCRATCH
   40          NODE2 = NODE2 + 1
               GO TO 70
   50          DO 60 NODE1 = NODE1START,NODE2START - 1
                  SCRATCH = DATARRAY(NODE1+1)
                  DATARRAY(NODE1+1)=DATARRAY(NODE1+1)+DATARRAY(NODE2+1)
                  DATARRAY(NODE2+1) = SCRATCH - DATARRAY(NODE2+1)
   60          NODE2 = NODE2 + 1
   70    TOGGLE = .NOT. TOGGLE
```

```
            NODE1START = NODE2
            GO TO 30
   80    BLOCKSIZE = BLOCKSIZE / 2
         RETURN
         END
*
         SUBROUTINE SCALE (NUMSAMPLS,DATARRAY)
*
C        WALSH TRANSFORM SCALING
*
         INTEGER I
         INTEGER*2 NUMSAMPLS
         INTEGER*4 DATARRAY(1)
         REAL SCALEFACTOR
*
         SCALEFACTOR = 1.0 /NUMSAMPLS
         DO 10 I = 1, NUMSAMPLS
           DATARRAY(I) = DATARRAY(I) * SCALEFACTOR
   10    CONTINUE
         RETURN
         END
```

LIST OF REFERENCES

1. Davis, D. S., "An Efficient and Versatile Method for Multiplexed Imaging," a paper in preparation.

2. McKenzie, R.H., *A Demonstration of the Use of Walsh Functions for Multiplexed Imaging,* Master's Thesis, Naval Postgraduate School, Monterey, California, December 1990.

3. Hecht, E., **Optics,** 2d ed., Addison-Wesley, 1987.

4. Beauchamp, K.G., **Applications of Walsh and Related Functions,** Academic Press, 1984.

5. Brigham, E.O., **The Fast Fourier Transform and Its Applications,"** Prentice-Hall, 1988.

6. Press, W.H., and others, **Numerical Recipes,** Cambridge University Press, 1986.

7. Elliot, D.F., and Rao, K.R., **Fast Transforms, Algorithms, Analyses, and Applications,** Academic Press, 1982.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Professor K.E. Woehler, Code PH/Wh Chairman, Department of Physics Naval Postgraduate School Monterey, California 93943-5000	1
4.	Assoc. Professor D.S. Davis, Code PH/Dv Department of Physics Naval Postgraduate School Monterey, California 93943-5000	2
5.	Assoc. Professor D.L. Walters, Code PH/We Department of Physics Naval Postgraduate School Monterey, California 93943-5000	1
6.	Department of Physics Library Naval Postgraduate School Monterey, California 93943-5000	2
7.	Commandant (G - ER) U.S. Coast Guard Headquarters 2100 Second ST. S.W. Washington, D.C., 20593-0001	1
8.	LT B.J. Musselman, USCG USCGC Diligence (WMEC-616) c/o Marine Safety Office Suite 500 272 N. Front St. Wilmington, North Carolina 28401-1817	2

Thesis
M98644
c.1

Musselman
A study of the diffraction behavior and resolution criteria for pattern recognition for a proposed multiplexed imaging technique.